

No. 1 i-Technology Magazine in the World



JDJ.SYS-CON.COM

VOL.11 ISSUE:5

— SEE PAGES 48-49 AND 60-61 FOR DETAILS —

<p><b>REAL-WORLD</b> <b>AJAX</b> ONE DAY SEMINAR</p> <p>June 5-6, 2006 New York, NY</p> <p><b>AJAXWORLD</b> CONFERENCE &amp; EXPO</p> <p>October 3-4, 2006 Santa Clara, CA</p> <p><a href="http://www.ajaxseminar.com">www.ajaxseminar.com</a></p>	<p><b>SOA</b> 10th International <b>WebServices</b> <b>Edge</b> conference+expo</p> <p>2006 <b>ENTERPRISE</b> <b>OPEN SOURCE</b> CONFERENCE+EXPO</p> <p>June 5-6, 2006 New York, NY</p> <p><a href="http://soaconference.sys-con.com">soaconference.sys-con.com</a></p>
--	---

# RIA<sup>WITH</sup> ADOBE FLEX 2 AND JAVA

((( Flash talks to POJO



RETAILERS PLEASE DISPLAY  
UNTIL JULY 31, 2006

\$5.99US \$6.99CAN

06>



0 09281 01751 6

## PLUS...

- ▶ Bringing Together Eclipse, WTP, Struts, and Hibernate
- ▶ J2EE/.NET Interoperability
- ▶ Using Self-Signed Certificates for Web Service Security



# Fireproof Your Code

## Prevent Java code fires with JProbe®

Constantly fighting Java code fires? Prevent infernos — before code moves into production — with award-winning JProbe from Quest Software.

Quickly pinpoint Java code hot spots with line-level analysis. Discover and debug memory leaks to dramatically improve performance. Automate the task of performing code analysis during off-peak hours. And release applications with confidence, knowing they have been fully tested. JProbe is the proactive solution that gives you higher levels of productivity and end user satisfaction.

Stop Java code performance flare ups — before they start. Improve code quality and increase application efficiency with JProbe.

---

Watch JProbe in action. View the new product demo at:

**[www.quest.com/JavaCode](http://www.quest.com/JavaCode)**

---

## Web 2.0

# 'Goes Mainstream'



Jeremy Geelan



### Editorial Board

Java EE Editor: **Yakov Fain**  
 Desktop Java Editor: **Joe Winchester**  
 Eclipse Editor: **Bill Dudney**  
 Enterprise Editor: **Ajit Sagar**  
 Java ME Editor: **Michael Yuan**  
 Back Page Editor: **Jason Bell**  
 Contributing Editor: **Calvin Austin**  
 Contributing Editor: **Rick Hightower**  
 Contributing Editor: **Tilak Mitra**  
 Founding Editor: **Sean Rhody**

### Production

Associate Art Director: **Tami Lima**  
 Executive Editor: **Nancy Valentine**  
 Research Editor: **Bahadır Karuv, PhD**

To submit a proposal for an article, go to  
<http://jdi.sys-con.com/main/proposal.htm>

### 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](mailto:subscribe@sys-con.com)

Cover Price: \$5.99/issue. Domestic: \$69.99/yr. (12 Issues)  
 Canada/Mexico: \$99.99/yr. Overseas: \$99.99/yr. (U.S. Banks or Money Orders) Back Issues: \$10/ea. International \$15/ea.

### Editorial Offices

SYS-CON Media, 135 Chestnut Ridge Rd., Montvale, NJ 07645  
 Telephone: 201 802-3000 Fax: 201 782-9638

Java Developer's Journal (ISSN#1087-6944) is published monthly (12 times a year) for \$69.99 by SYS-CON Publications, Inc., 135 Chestnut Ridge Road, Montvale, NJ 07645. Periodicals postage rates are paid at Montvale, NJ 07645 and additional mailing offices. Postmaster: Send address changes to: Java Developer's Journal, SYS-CON Publications, Inc., 135 Chestnut Ridge Road, Montvale, NJ 07645.

### ©Copyright

Copyright © 2006 by SYS-CON Publications, Inc. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system, without written permission. For promotional reprints, contact reprint coordinator Megan Mussa, [megan@sys-con.com](mailto:megan@sys-con.com). SYS-CON Media and SYS-CON Publications, Inc., reserve the right to revise, republish and authorize its readers to use the articles submitted for publication.

Worldwide Newsstand Distribution  
 Curtis Circulation Company, New Milford, NJ

For List Rental Information:

Kevin Collopy: 845 731-2684, [kevin.collopy@edithroman.com](mailto:kevin.collopy@edithroman.com)  
 Frank Cipolla: 845 731-3832, [frank.cipolla@epostdirect.com](mailto:frank.cipolla@epostdirect.com)

Newsstand Distribution Consultant  
 Brian J. Gregory/Gregory Associates/W.R.D.S.  
 732 607-9941, [BJGAssociates@cs.com](mailto:BJGAssociates@cs.com)

Java and Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. SYS-CON Publications, Inc., is independent of Sun Microsystems, Inc. All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies.



**W**hen newsstands throughout America on Monday, March 27, started displaying the April 3 issue of *Newsweek* with its cover story about "Web 2.0" – "Putting the 'We' in Web" – it seems to me that we have reached one of Malcolm Gladwell's now-famous *Tipping Points*.

In the *Newsweek* piece, written by the magazine's senior editor Steven Levy and its Silicon Valley correspondent Brad Stone, the authors report how, "less than a decade ago, when we were first getting used to the idea of an Internet, people described the act of going online as venturing into some foreign realm called cyberspace." But that metaphor, the authors suggest, no longer applies: "MySpace, Flickr, and all the other newcomers aren't places to go, but things to do, ways to express ourselves means to connect with others and extend our own horizons."



"Cyberspace was somewhere else," they write. Whereas "*The Web is where we live.*" [My emphasis.]

What's the tipping point, you ask. Well, for me – and for plenty of others, it seems – what's special/unusual is that *Newsweek*, a mainstream publication, deems "Web 2.0" worthy not just of an in-depth article but also of an in-depth article that it puts on its cover. It means that the notion of a "live Web" is getting prime time, along with the terminology of Web 2.0 like "tagging," "mash-ups," "collective intelligence," and "social media."

Web 2.0, make no mistake about it, is going mainstream.

For those who think that one weakness of the *Newsweek* piece is its title, MSNBC has come to the rescue by repurposing it under the – in my view,

far sharper – title "The New Wisdom of the Web." This is a much more powerful rallying cry and I, for one, should have much preferred to see it used as *Newsweek's* front-cover headline, but such is the way of the world. Maybe the editors at MSNBC "get it" better than those at *Newsweek*.

I can't wait for *BusinessWeek* to address the same story: they'll get the title spot-on, just watch!!

*Newsweek's* two authors triangulated on James Surowiecki, whom they quote, so his "Wisdom of Crowds" notion is included in the article. ("It's clear that the Web is structurally congenial to the wisdom of crowds," he's quoted as saying.) But my instincts tell me that, for all that Steven Levy and Brad Stone are to be thanked and congratulated for producing the article, they don't necessarily – yet, anyway – fully comprehend the technologies of "social computing" like RSS. And they certainly don't have a clue what AJAX actually is. (It is hastily referenced in a single sentence buried right in the middle of

the piece.)

Again, all credit to *Newsweek*, though, for the resonant final paragraph:

*Less than a decade ago, when we were first getting used to the idea of an Internet, people described the act of going online as venturing into some foreign realm called cyberspace. But that metaphor no longer applies. MySpace, Flickr and all the other newcomers aren't places to go, but things to do, ways to express yourself, means to connect with others and extend your own horizons. Cyberspace was somewhere else. The Web is where we live.*

With these five sentences, given their appearance in one of the world's most widely read news magazines, "Web 2.0" has truly arrived in Main Street. ●

**Jeremy Geelan** is group publisher of SYS-CON Media and is responsible for the development of new titles and technology portals for the firm. He regularly represents SYS-CON at conferences and trade shows, speaking to technology audiences both in North America and overseas.

[jeremy@sys-con.com](mailto:jeremy@sys-con.com)

DESKTOP

CORE

ENTERPRISE

HOME



# Streamline Web services

Hook up with MapForce® 2006, and build  
Web services without writing any code.

## Spied in MapForce 2006:

- Drag-and-drop Web services implementation
- Advanced flat file parsing and integration
- Project-wide code generation
- Embedding in your applications via OLE / ActiveX

Altova® MapForce, the tool awarded for easy integration of XML, database, text, and EDI file formats, now also lets you implement Web services in a visual way. Simply drag connecting lines from information sources to targets and drop in data processing functions. MapForce converts data on-the-fly and auto-generates data mapping code in XSLT 1.0/2.0, XQuery, Java, C++, or C# for use in your data integration and Web services applications. Give your data direction!

**Download MapForce® 2006 today: [www.altova.com](http://www.altova.com)**

Also available in the Altova XML Suite.

Visit Altova  
at JavaOne  
booth # 626



# JDJ contents

## JDJ Cover Story

# RIA<sup>WITH</sup> ADOBE FLEX 2 AND JAVA

(((Flash talks to POJO

by Yakov Fain, Victor Rasputnis,  
and Anatole Tartakovsky

# 22

FROM THE GROUP PUBLISHER

### Web 2.0 'Goes Mainstream'

by Jeremy Geelan

.....3

ENTERPRISE VIEWPOINT

### Interesting Times in the Java Enterprise

by Rick Hightower

.....6

JAVA

### Java Development Managers Stress the Need to Improve Software Quality

*Lack of attention early in the development cycle  
negatively impacts the code's entire lifecycle*

by Nigel Cheshire

.....10

INTEROPERABILITY

### j2EE/.NET Interoperability

*A look at managing complex interoperating  
systems*

by Laurence Moroney

.....30

Q & A

### Blended Development

*Interview with Blake Connell, Director of Product  
Marketing for WebLogic Server, BEA Systems*

by Roger Strukhoff

.....36

VIRTUAL SOCKETS

### Using Virtual Sockets to Fix Software Broken by Firewalls

*Without violating security*

by Ron Sigal and Tom Elrod

.....38

PROJECTS

### Maven: A Different Way of Looking at Software Development

*A real-time engineering framework*

by Jason van Zyl

.....42

DESKTOP JAVA VIEWPOINT

### Whatifitis: 'To Dwell in the Future and Forget About Today'

by Joe Winchester

.....46

SECURITY

### Using Self-Signed Certificates for Web Service Security

*How to compete with trusted certificate authorities*

by Michael J. Remijan

.....54

JDJ EDITORS

### Meet the JDJ Editors

.....58

JSR WATCH

### 2006 JavaOne JSR Itineraries

by Onno Kluyt

.....62

## Features



# 14

### Bringing Together Eclipse, WTP, Struts, and Hibernate

by Boris Minkin



# 50

### Visualizing Data Using Java3D

by Valor Dodd

President and CEO:

Fuat Kircaali fuat@sys-con.com

Group Publisher:

Jeremy Geelan jeremy@sys-con.com

### Advertising

Senior Vice President, Sales and Marketing:

Carmen Gonzalez carmen@sys-con.com

Vice President, Sales and Marketing:

Miles Silverman miles@sys-con.com

Advertising Sales Director:

Robyn Forma robyn@sys-con.com

Advertising Sales Manager:

Megan Mussa megan@sys-con.com

Associate Sales Manager:

Kerry Mealia kerry@sys-con.com

### Editorial

Executive Editor:

Nancy Valentine nancy@sys-con.com

### Production

Lead Designer:

Tami Lima tami@sys-con.com

Art Director:

Alex Botero alex@sys-con.com

Associate Art Directors:

Abraham Addo abraham@sys-con.com

Louis F. Cuffari louis@sys-con.com

Assistant Art Director:

Andrea Boden andrea@sys-con.com

### Web Services

Information Systems Consultant:

Robert Diamond robert@sys-con.com

Web Designers:

Stephen Kilmurray stephen@sys-con.com

Wayne Uffelman wayne@sys-con.com

### Accounting

Financial Analyst:

Joan LaRose joan@sys-con.com

Accounts Payable:

Betty White betty@sys-con.com

Accounts Receivable:

Gail Naples gailn@sys-con.com

### SYS-CON Events

President, SYS-CON Events:

Grisha Davida grisha@sys-con.com

National Sales Manager:

Jim Hanchrow jimh@sys-con.com

### Customer Relations

Circulation Service Coordinator:

Edna Earle Russell edna@sys-con.com

IDJ Store Manager:

Brunilda Staropoli bruni@sys-con.com



**Rick Hightower**  
Enterprise Editor

# Interesting Times in the Java Enterprise

Robert F. Kennedy once said, "There is a Chinese curse which says, 'May he live in interesting times.'" The enterprise Java space is "interesting."

Not too long ago, folks like Bruce Tate, Gavin King, and Rod Johnson were pushing lightweight frameworks such as Spring and Hibernate, and there is still a lot of true innovation going on with AspectJ, Spring, Hibernate, WebWork, JBoss (method invocation handlers), and more. This lightweight POJO revolution shook the enterprise Java world.

Having endured building applications with EJB 2.x and Struts, using Spring and Hibernate was like a breath of fresh air. Development was easier and less time was spent working around the limitations of the platform.

J2EE has good ideas, which inspired a lot of additional ideas. This evolution led to innovative and productive practices outside of the JCP.

The JCP has had some very good JSRs, but you have to admit there have been some real stinkers.

EJB in general, except as a learning experience, is generally viewed as a failure. The real problem was not just EJB but the misapplications of EJB. This was widespread as it was promoted with the J2EE Blueprint, not to mention the misapplication of JTA, and more features of J2EE. Not to say that applications can't benefit from JTA and EJB, it's just that many, many Web applications don't need them.

EJB 3.0 is much better than EJB 2.x. If you compare EJB3 to an older version of EJB, EJB3 is a boon; however, if you compare EJB3 to Spring and Hibernate, it stinks.

The related *OR (Object Relation) Persistent API* does not have a criteria API specified; any persistent API that does not define a criteria API is not finished.

The AOP support in EJB3 is broken. EJB3 has a method interceptor, but no pointcuts. In addition, the method interceptors are declared and imported with class-level annotations. This effectively tightly couples the class to the method interceptors that decorate it (Can you smell the bad odor?).

Rod Johnson mentioned these

same problems about  
EJB3 Method

Interceptors at  
a recent Java  
enterprise  
conference  
(in his talk  
"Are We  
There Yet")  
and went on  
to mention  
many limita-  
tions on the

@Resource style of  
DI, the absence of Facto-

ryBeans, post processors, Constructor Injection, lists/maps, and a lot of the features Spring developers know and love are just missing. The EJB3 JSR members did not look at any of the prior art in this space and created their own limited version of what was already available.

I've heard some call EJB3 a dumb-down version of what is available by using Spring and Hibernate. "EJB3 is Spring and Hibernate's stupid cousin" is frequently echoed.

After three years of deliberation, the JCPs delivered EJB3, which is inferior to de facto standards. Many parts of EJB3 are a big step backward from Spring, and, to many, EJB3 is broken. As Bruce Tate says about EJB3: "Don't make me eat the elephant again."



**Rick Hightower** serves as chief technology officer for ArcMind Inc (<http://www.arc-mind.com>). He is author of several books on Java web development. He has been working with J2EE since the very early days and lately has been working mostly with Maven, Spring, JSF, and Hibernate. Read his blog at <http://jroller.com/page/RickHigh>

[rhightower@arc-mind.com](mailto:rhightower@arc-mind.com)

—continued on page 8

## Innovations by InterSystems



*World's fastest object database*



*Objects stored directly in a multidimensional engine*



*Massive scalability on minimal hardware*



*Object-relational mapping eliminated*

## Object Database That Runs SQL Faster Than Relational Databases.

Caché is the world's fastest object database. Plus, it runs SQL queries up to five times faster than relational databases. Its innovative multidimensional engine combines advanced object technology and robust SQL, eliminating object-relational mapping. *Which enables Java programmers to create applications in 40% less time.*

We back these claims with this money-back guarantee: Buy Caché for new application development, and for up to one year you can return the license for a full refund if you are unhappy for any reason.

Caché is available for Unix, Linux, Windows, Mac OS X, and OpenVMS – and it's deployed on more than 100,000 systems ranging from two to over 50,000 users. We are InterSystems, a global software company with a 28-year track record of innovations that enrich applications.



Try an innovative database for free. Get your copy at our JavaOne booth #522, or download it at [www.InterSystems.com/JavaOne2006P](http://www.InterSystems.com/JavaOne2006P)

\* Read about our money-back guarantee at the web page shown above.

© 2006 InterSystems Corporation. All rights reserved. InterSystems Caché is a registered trademark of InterSystems Corporation. 4-06 CachéInnoJavaOneJDJ



—continued from page 6

It's not just the persistent API and the AOP support that's broken in EJB3, it's also the random use of annotations, another misguided effort. The idea of annotations is good. The implementation of the annotations ruins some of the principles of the POJO model; namely, it ties your Java classes through a compile-time dependency to the standard API you're using and to any value-add annotations the vendor supports. Now why would vendors like this approach? Hmmm...I wonder. (*Hint: Follow the money!*)

In that question lies the real problem with the JCP. The JCP is heavily influenced by vendors that have "business need(s) or corporate agenda(s)." Parts of the enterprise Java community is innovative, parts stink, but there are many parts.

Strangely enough, RoR, which is currently being championed by Bruce Tate among others, is a safe haven for Java developers who are sick of vendor-driven APIs. In short, vendor-centric

There are good Java competitors to RoR, such as Rife and Seam (and many others). The way the Java competitors of RoR do things is better than RoR, and it's Java so you get code completion, refactoring support, debugging, and more. Java can attack the space that RoR addresses quite well, in fact, much better than RoR. As Gavin King recently wrote me: "...Seam is (not) exactly focused on the same kinds of problems that RoR is targeted at, but I would say that it does compete very nicely in that space."

The reality is, we need more independent voices in the JCP. To be successful, it has to become less vendor-driven.

At least we now have independent folks on the JCP like Hani Suleiman. Hani is a self-proclaimed independent voice of the JCP. Hani, a very popular blogger, petitioned Java developers to vote him into the JCP Executive Committee for J2SE/J2EE.

"If you're a JCP member, vote for me...! ... I'm the only nominee who is motivated purely by improving Java.

read Hani's blog, he frequently speaks out against AOP, POJOs, and lightweight frameworks, not to mention bashing competitors like JBoss, etc. It also appears there has never been an EJB he didn't like. And, this is our "independent voice." Strange times indeed!

The question to Hani, which echos a popular U.S. TV commercial: "*But sir, aren't you the man?*"

Even so, many feel that Hani will defend the common developer against vendor's interests and that he will bring a new perspective to the JCP, while ignoring that Hani, pleasant chap though he is, is hardly an independent voice and often lashes out against innovation while supporting a broken specification like EJB. One thing for certain, Hani never pulls his punches.

Hani recently told me: "The JCP isn't really a technical body; it's a lot more about process and IP issues." So a big part of the JCP's work is about protecting the individual vendor's IP issues. Where is the focus on making developer's productive like RoR's focus? By and large, RoR does not beat out the

## “Having endured building applications with EJB 2.x and Struts, using Spring and Hibernate was like a breath of fresh air”

JSRs, Struts, and EJB have driven many a developer, who just wanted to get things done, to RoR.

My feelings on RoR is, been there done that, no thanks. Don't get me wrong, RoR has a lot of good ideas, but bad tool support. For me and many other Java developers, scripting languages are a step backward for large apps.

Geert Bevin, the creator of Rife, sums up my thoughts on the subject nicely in a recent e-mail he sent: "RoR is one of the best things that could happen to the Java community because at least alternative approaches and meta programming are now getting the credibility they deserve. Technologically, though, I think that RoR is nothing special." An issue with Java is a wealth of riches, so much wealth that you can get lost. Conversely, RoR seems like a one-trick pony. Choice is a blessing and a curse.

Everyone else is there out of some business need or corporate agenda. *Stick it to the man!*" —Hani Suleiman

Of course, many of us are very happy to have such an independent voice, an independent developer who doesn't have any vested interest in any of the JSRs. But wait...

Hani also did an interview in which he states: "I am the CTO of Formicary, which ... (has a) portal product, Epix (a JSR-168 based Portlet Portal), and alongside all that I also work for IronFlare, which makes the Orion application server, and through that I represent them on a number of expert groups and that is the real work that I do."

Hani works for companies that produce a portal server and an application server that support EJB. IronFlare was contracted by Oracle to develop their J2EE application server (the second one, the good one). Hani is a vendor. If you

Java community in tools and ideas, but on their focus on developer productivity. Contrast this to the JCP's focus on vendor profits. We need to focus.

We do live in interesting times in the Enterprise Java space. We have RoR-inspired frameworks and vendor-driven JSRs. We have Bruce Tate, a former promoter of lightweight Java frameworks, become an advocate for RoR. We have Hani, one of the chief opponents of lightweight POJO-based frameworks, get elected to the JCP Executive Committee.

On the other hand, we have great frameworks and ideas such as POJOs, domain-driven design, Spring, Hibernate, Rife continuations, Rife meta programming, JSF, Facelets, Seam, and AspectJ. It is time to promote true innovation wherever we find it: JCP or innovative projects. Avoid standard APIs and frameworks that don't make sense. ☺



Some  
of your  
important  
content  
not  
rendering?



## get WebRenderer™

### Java Browser SDK

#### Desktop Edition

Embed WebRenderer into your Java client side application interface to display rich web content.

- ✓ Swing and AWT support
- ✓ Extensive DOM
- ✓ Plugin support
- ✓ Powerful API
- ✓ HTTP, SSL, TLS
- ✓ JavaScript
- ✓ Complete Event model

#### Server Edition

Automate Server Side web content reporting, printing and processing with WebRenderer Server Edition.

- ✓ Headless server optimized
- ✓ J2EE support
- ✓ No AWT or Swing dependencies
- ✓ Complete DOM support
- ✓ HTML to thumbnail image
- ✓ Print support
- ✓ Standards compliant rendering

To download a free 30 day trial visit [www.webrenderer.com](http://www.webrenderer.com)

# Java Development Managers Stress the Need to Improve Software Quality

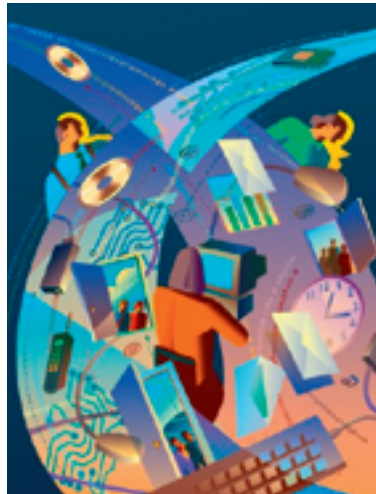
by Nigel Cheshire

*Lack of attention early in the development cycle negatively impacts the code's entire lifecycle*

**W**e've all experienced it – the “get it out the door” mentality that seems to be the driving force behind many software application deliveries – a prime example of the software industry's immaturity that favors completion over quality, and an end user's preference for hot new features over stable, reliable systems. Deferring the QA process is an expensive way to operate and corporations are taking a financial hit for these software errors. According to the Washington, D.C., National Institute of Standards and Testing (NIST), software errors cost the U.S. economy \$60 billion per year. This report was issued back in 2002 and, since then, the software industry has done little to improve the situation.

Why are software bugs so pervasive? We don't accept these same poor standards in other industries, so what makes software exempt? Some say it's the inherent complexity of systems and software that are the primary causes of application failure, or that poorly defined system requirements and unrealistic schedules are at fault. Others blame it on developer apathy. Most of the time, however, software failure is the direct result of poor quality code, brought about by a lack of standards and metrics in the development phase.

A Clear Horizons market research study, commissioned by Enerjy Software, identified key code quality issues that impact software development managers. First, quality tracking begins too late, if at all, to be effective. According to the study, 40 percent of Java software development managers surveyed have allotted time for checking code quality, but only in less than half of their projects. In projects where quality is being measured, nearly 30 percent did not measure code quality until the project was more than half completed. This lack of attention early on in the development cycle has a rolling negative impact for the entire life cycle of the code.



## Focusing on Code Quality

The study found that most Java software development managers view code reviews, the traditional method of improving code quality, as inefficient and a source of frustration. With all the competing demands on their limited development resources, it often falls by the wayside. As a result, the act of fixing a bug may be the only time their code is actually reviewed. By that time, it is too late.

Development managers are demanding a change from the “fire fighting” mentality prevalent in the industry. However, this change needs to come from IT executives who must take the lead in establishing best practices that focus on quality across the entire software development lifecycle – from requirements gathering to deployment and beyond. Applying best practices to help achieve software quality must become a mindset that is filtered from the top down and includes:

- Requirements definitions
- Unit testing
- Coding standards adherence
- Change and version control
- Quality assurance testing
- Software maintenance post-release

It is imperative that quality initiatives begin before the first line of code is written. A key reason for software project overruns is that the time needed for the QA phase is usually grossly underestimated. This is because the QA phase should be used for functional testing, i.e., checking that the application does what it is supposed to do. Instead, too much time is being consumed in the QA phase, finding and fixing bugs that should have been fixed in the development phase. Errors need to be identified and prevented from progressing through the development cycle, because, as everyone knows, it's more costly to fix bugs later in the development process than at the beginning.

Early in the development cycle, developers need to define coding standards, adhere to them, and ensure that all code is unit tested. By applying coding standards and unit testing at the development stage, many potential bugs will never occur. During the quality assurance phase, standardized testing tools, suites, scripts, reference data, and metrics that have undergone a rigorous certification process can improve software integrity even more.

A focus on code quality enables development teams to optimize the quality of an application before it reaches production. Such a QA process will help to make testing less costly to perform, as well as provide a consistent method for determining when testing is sufficient. The focus on code quality throughout a Java application's life cycle will enable development teams to fix bugs and add new features within schedule while improving overall product quality.

## Who's Responsible?

While software developers are responsible for maintaining or improving bad code, it is the development managers who are directly responsible for



**Nigel Cheshire** is CEO of Enerjy Software, a division of Teamstudio Inc. He oversees product strategy and has been driving the company's growth since he founded it in 1996.

Prior to founding Teamstudio, Inc., Cheshire was co-founder and principal of Ives & Company, a CRM solutions consultancy. Nigel holds a Bachelor of Science in computer science from the University of Teesside, England.



# Perforce Software Configuration Management

# Powerful

# and scalable.

Meet Perforce Software at JavaOne Booth #1040

Ask for a demo of the IBM-Certified Perforce WebSphere Studio Plug-in, P4WSAD  
Also a member of the Eclipse Foundation

[Fast]

[Scalable]

[Distributed]

Perforce tracks and manages source code and digital asset development, giving you maximum control with minimum interference.

Perforce scales to **thousands of concurrent users**. It manages code bases of **more than a million files**. It processes **terabytes of data** - from source code to documents, web content, and image files.

Perforce is fast, industrial-strength SCM. Any team size, any amount of data - Perforce handles it.

**PERFORCE**  
SOFTWARE

Download a free copy of Perforce, no questions asked, from [www.perforce.com](http://www.perforce.com). Free technical support is available throughout your evaluation.

All trademarks used herein are either the trademarks or registered trademarks of their respective owners.

code quality. Their jobs are to deliver code that works and, with the same resources, ensure that critical bugs are identified and fixed in a timely manner – an almost impossible balancing act.

In fact, buggy code is the “Sword of Damocles” for all software development managers. It’s what hangs over their heads and keeps them awake at night. According to the aforementioned study, the number one complaint of Java software development managers is their inability to minimize software bugs in code before it is released. This study, completed in 2005, was conducted on a sampling of 216 Java software development managers who manage five or more Java developers.

Tasked with delivering quality code, how can Java development managers best facilitate quality processes? What are their primary challenges? What tools do they need to accomplish their

management training and technical training on par with their subordinates; however, this is not enough. Today’s management and lead developers are stuck in a cycle of poor code and bug fixing, without the corporate backing required to identify issues and initiate software development best practices.

Another issue presented in the report is the lack of quality metrics available to Java development managers. A majority of the Java development managers in the Clear Horizons study reported frustration with developers not following Java best practices or coding standards, or producing “off-spec” code – yet they had no way to track this information via quality metrics. Without these quality metrics, development managers have no idea who is creating the bad code, or what percentage of that code adheres to standards or is being unit tested. As development managers, their primary

sis placed on the quality of a team’s deliverables at all points within the development cycle.

Java software development teams want to create quality code. Their apathy and resistance to change is often due to a stagnant development environment, over which they have little control. Teaching them about the benefits of best practices, such as the adherence to coding standards and thorough unit testing, is not enough. Most developers have already been exposed to these methodologies. Development managers, who are ultimately responsible for code quality, often feel they have their hands tied. Compressed schedules, unknown developer performance – especially with new hires and novice developers, the absence of best practices and controls, and a bug count that sidetracks resources from competitive enhancements, all conspire to keep them from sleeping at night.

“While software developers are responsible for maintaining or improving bad code, it’s the development managers who are directly responsible for code quality”

objectives? It makes sense that a focus on quality is essential for the success of any software application.

### Stressing the Importance of Code Quality in the Development Environment

Though development managers take the heat for missed deadlines, wasted development resources, and poor application quality, software development organizations have primarily focused on providing developers with the tools, infrastructure, and training for developing software applications.

While it is easy to blame software errors on developers – sloppy coding techniques, arrogance, and an insistence upon the freedom to create – it may be the development environment itself that is the problem. Typically, development managers have been promoted from within the development ranks due to their outstanding technical abilities and may not have had previous management training or experience. Often, they receive generic

objective is to prevent or minimize bugs. Without an in-depth understanding of the importance of code quality relative to individual developer performance, they are flying blind.

Most developers prefer to work in a structured development environment that supports best practices, because it offers relief from “fire fighting” and damage control. By introducing best practices into an organization’s development environment, developers will find errors early, streamline the build processes, reduce overtime, and become clear on their roles within the organization.

### Improving Tomorrow’s Software Quality

With maintenance costs exceeding 90 percent of organizations’ IT budgets, it is imperative that development managers work to transform their team’s behaviors from reactive to proactive: instilling best practice tactics that focus on superior code quality and increased application development excellence. To do this, there must be greater empha-

In order for best practices to take hold, IT management must implement these best practices across the development team and promote code quality as a mindset from the top down. Development managers have to promote coding standards initiatives that are easily embraced by all developers and enact processes and procedures, such as source code control systems, that streamline processes, and unit testing, to eliminate many potential bugs before the software is released to Quality Assurance.

What Java development managers need most of all is a quality measurement system that institutes code quality metrics and helps them track and train developers, so that they can meet deadlines with high quality software that conforms to internal standards and exceeds customer requirements. Such a system would give Java development managers a greater sense of control over their development team and the quality of code that they produce. ☛

# EMPOWERING DEVELOPERS WORLDWIDE



## NetBeans

NetBeans Platform:  
Your Business Solution

- ✓ **Open Standards, 100% Java**
- ✓ **Cross-Platform, Modular, Scalable, Extensible**
- ✓ **Application Prototyping Utilizing NetBeans GUI Builder Matisse**
- ✓ **Free & Open Source Application Development Framework**

Find out more at  
[www.NetBeans.org](http://www.NetBeans.org)



# Bringing Together Eclipse, WTP, Struts, and Hibernate

*Improve application maintainability, code reusability, and code clarity*

by Boris Minkin

In the article “Creating Web Applications with the Eclipse WTP” (<http://jdj.sys-con.com/read/152270.htm>), we created a Web application using Eclipse Web Tools Project, the Tomcat application server, and the MySQL database server. That application (DBTest) was good, however, it had some limitations:

1. Java Server Pages (JSP) names were hard-coded inside the servlet code
2. SQL was also hard-coded in the command classes

Fortunately, two interesting solutions can address these problems. The first problem can be addressed using the Open Source Struts framework, which separates an application's Model, View, and Controller by mappings the actions of the model to view components (such as JSPs) in a simple configuration file.

The second problem can be addressed by using one of the frameworks providing persistence between the Java world and the relational database world. Hibernate framework provides a powerful high-performance mapping engine between the objects and database tables. The following technology is used in this article:

1. J2SE 5.0 JRE: <http://java.sun.com/j2se>
2. Eclipse 3.1: [www.eclipse.org](http://www.eclipse.org)
3. WTP 1.0: [www.eclipse.org/webtools](http://www.eclipse.org/webtools)
4. Tomcat 5.0: <http://jakarta.apache.org/tomcat/>
5. MySQL 4.0.25: <http://www.mysql.com>
6. MySQL Connector/J driver 3.1: <http://www.mysql.com/products/connector/j/>
7. Struts 1.1: <http://struts.apache.org>
8. Hibernate 3... <http://www.hibernate.org>

## Application Overview

Let's recap what we did last time. It was a basic Web application that implemented the following use cases:

- Customers have to register on the site to place orders
- Customers can place orders
- Customers can view their orders
- Admin can list all registered customers

The system was implemented using the generic servlet/JSP programming model and the MySQL database, with Tomcat application server. Two classes, Customer and Order, represented the domain model of the system (see Figure 1).

Two corresponding database tables CUSTOMERS and ORDERS were created to represent the data that were held in these objects. We also created four database command classes that were responsible for performing the aforementioned use cases, and four servlets acting as controllers to gather input from the user, invoke these commands, and forward the response to the appropriate JSP. A special class CommandExecutor was responsible for handling the database connections using Tomcat connection pools.

## Adding Struts Support

Import the DBTest.WAR file (<http://java.sys-con.com/read/152270.htm>) into your Eclipse workspace using the File-Import option and selecting the WAR file item to import. That will work perfectly if the DBTest project isn't already in your workspace. If DBTest project is already in your workspace, just copy it to preserve the existing project by right-clicking on the project in the Navigator view and selecting Copy and then Paste when you'll be prompted for the new project name (we'll select DBTestStruts as the new name), so that the existing project isn't clobbered. Now to add Struts support, we have to copy the following files into the WEB-INF\lib folder: struts.jar, commons-lang.jar, commons-collections.jar, commons-beanutils.jar, commons-validator.jar, commons-logging.jar, commons-digester.jar, commons-fileupload.jar.

All these files can be downloaded from the Struts Web site and contain the Struts framework along with corresponding Apache Commons packages necessary for handling features such as internationalization, collection operations, utilities, validation, logging, digester, and file upload operations. They are all supporting parts for Struts. We won't use all of these features here, but Struts relies on many of them (e.g., the digester part is heavily used when parsing a Struts configuration file), and they may become handy later when services such as logging and file uploading are required.

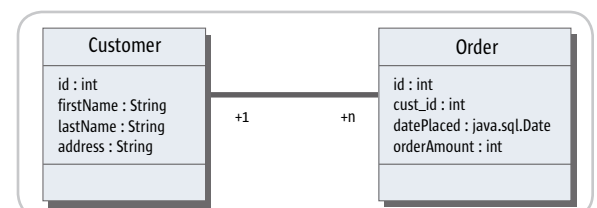


Figure 1



**Boris Minkin** is a Divisional Vice President of a major financial corporation. He has more than 12 years of experience working in various areas of information technology and financial services. Boris is currently pursuing his Masters degree at Stevens Institute of Technology, New Jersey. His professional interests are in the Internet technology, service-oriented architecture, enterprise application architecture, multi-platform distributed applications, and relational database design.

[bm@panix.com](mailto:bm@panix.com)

# #1 Rated AJAX and Rich Internet Application toolkit\* TIBCO General Interface

**Build 100% Pure Browser Rich Internet Apps with TIBCO General Interface™**

“TIBCO General Interface is a friendly, capable toolkit for building sophisticated JavaScript Web applications that run in a browser.”



Download today  
at <http://www.tibco.com/mk/gi>

 **TIBCO®**  
The Power of Now®



Figure 2

So add the following files to the WEB-INF folder: struts-config.xml, struts-bean.tld, struts-html.tld, struts-logic.tld, struts-nested.tld, struts-template.tld, struts-tiles.tld.

The most important of these files is struts-config.xml, which is a main configuration file for the Struts framework. It contains definitions of all the action mappings, data sources, plug-ins, etc. See the sample one in Listing 1.

TLD files are Struts tag library definition files that can be used inside of JSPs to do various useful operations such as HTML rendering, logic handling, or Tiles support. They can be obtained from the Struts 1.1 distribution.

The next thing we need to do is modify our Web Deployment Descriptor (web.xml) to specify the location of the Struts configuration servlet and corresponding parameters. The code snippet from Listing 2 should be added to web.xml.

Tags from Listing 2 define the location of the Action Servlet, which is the Struts primary controller responsible for handling the lifecycle of the actions and mapping them to forwards, which are objects that are returned by an action and have two fields: a name and a path (typically the URL of a JSP file). The location of the struts-config.xml file is specified here, as well as the parameters for debugging and validation. The servlet is loaded at startup and the order of its load is 1 (first servlet to be loaded). The servlet gets invoked whenever a \*.do pattern is detected in the invoking URL.

Now we have to convert our existing servlet classes into action classes and define appropriate mappings for them in struts-config.xml. To simplify this process, we'll provide an abstract superclass for all of our actions (see Listing 3) (Listings 3-15 and additional source code can be downloaded from the online version of this article at <http://jdj.sys-con.com>.)

In this class, we provide implementation of the "execute" method that's invoked by default on the action by the Struts 1.1 framework. It handles the logic in its performAction()

method and then forwards to either success or failure processors depending on whether an exception has been thrown or not. Correspondingly, mappings for both success and failure will have to be defined in the Struts configuration file (struts-config.xml) for every action.

Creating concrete actions is easy. We can use the Eclipse wizard to create action classes. Make sure that AbstractAction is selected as a superclass and the "Inherited abstract methods" box is checked (see Figure 2).

The class CreateCustomerAction with performAction() method will be automatically generated. Copy the contents of the CreateCustomerServlet doGet() method (see previous article at <http://java.sys-con.com/read/152270.htm> -) and paste it into performAction() with the modifications shown in Listing 4.

As one can see, the only difference between non-Struts code and Struts code is that instead of the following code:

```
RequestDispatcher rd = getServletContext().getRequestDispatcher("/customer_created.jsp");
rd.forward(request, response);
```

the following (simpler one) is used:

```
return mapping.findForward("customer_created");
```

We don't have to hard-code the name of the JSP inside our code any more. Instead, we use the "customer\_created" reference, which will be resolved in the struts configuration file. Inside the <action-mappings> tags, we'll add the fragment in Listing 5.

In this example, /CreateCustomer would be the URI to invoke the action. Two forwards are defined – "customer\_created" – which, incidentally, points to "customer\_created.jsp" and "failure," which points to "failure.jsp" where errors can be displayed. It's useful to have a common error page for the application, and we'll create one right now (see Listing 6).

In this file, we use the Struts HTML tag library to display the errors captured.

In a similar manner, we'll convert other servlets into Struts actions. Don't forget to change the URLs in the index.html file and other JSP files and add a suffix ".do" to the action invocations. Changing references from DBTest to DBTestStruts is also necessary to deploy the new application in co-existence with the old one in the Tomcat server. Also change the display name in web.xml from DBTest to DBTestStruts.

Remove the old servlet definitions from the DBTest application source code and Web Deployment Descriptor – all we need there is just the definition of our Actions and the Action servlet. To delete the "servlet" package, simply right-click on the "servlet" package and select "Delete," then answer, "Yes," when prompted for confirmation.

To deploy the new application to Tomcat open its console <http://localhost:8080/manager/html> and deploy a new WAR file. Make sure that DBTest.xml is copied into DBTestStruts.xml and all the references to DBTest are changed to DBTestStruts in it.

Another issue, however, is that in the original solution's SQL was hard-coded directly inside the command classes. This will be addressed in the next section by the popular Hibernate framework, which supports persistence between Java and relational databases.





**RCP Developer™** is created by the experts who brought you the popular book  
*Eclipse: Building Commercial Quality Plugins* — Eric Clayberg & Dan Rubel

**Java on the desktop is back!**

# RCP Developer™



Instantiations RCP Developer harnesses the power of the Eclipse Rich Client Platform. Quickly create native desktop applications that are rich, cross-platform, and high-performance. The vast expertise of the Eclipse developer community and the experience of Instantiations lets you...

## STAND ON THE SHOULDERS OF GIANTS

### Create blazing fast rich Java clients.

Tap into the ultra high performance Eclipse graphical user interface libraries and ensure a native look and feel.

### Build rich Java client applications faster.

Focus on creating business specific functionality rather than reinventing the low-level logic required for applications to run.

### Deliver a more consistent user experience.

Tame the most demanding user interface requirements and reliably deliver a rich and consistent user experience across all applications.

### Fully exploit the Eclipse Rich Client Platform.

"The Eclipse Rich Client Platform is the leading framework for creating Java client applications. RCP Developer is the first product to bring comprehensive application construction, GUI testing and packaging of rich-client applications to Eclipse RCP"

—Mike Milinkovich, executive director of the Eclipse foundation

### RCP Developer™

#### SWT Designer™



**Design.** Quickly create views, editors, perspectives, and preference pages for your applications. Intuitive visual design and access to the high performance SWT GUI framework of Eclipse RCP.

#### WindowTester™



**Test.** Automatically test the GUIs that comprise Eclipse RCP applications. Minimize the need to hand-code GUI test cases. Automated recording, test generation, assertion coding and playback facilities.

#### RCP Packager™



**Deploy.** Streamline build and deployment for Eclipse RCP applications. Assemble application elements into a single package. Quickly create and maintain high-quality Eclipse RCP installations.



[www.instantiations.com](http://www.instantiations.com) 1-800-808-3737

*The leader in Eclipse RCP development tools*



© Copyright 2006 Instantiations, Inc. RCP Developer, SWT Designer, WindowTester and RCP Packager are trademarks of Instantiations. All other trademarks mentioned are

the property of their respective owners.



**Download a risk-free trial copy:**

[www.instantiations.com/rcpdeveloper](http://www.instantiations.com/rcpdeveloper)

## Adding Hibernate Support

The Hibernate framework helps you in the following areas:

1. Object-relational mapping. It allows you to seamlessly map Java objects and the relationships between various classes into database tables and the relationships among them. This is done using XML configuration files and saves a lot of the time that developers spend coding custom SQL queries and constructing objects from JDBC result sets.
2. Connection management. Reusing existing database connections is one of the most important efficiency tuning mechanisms.
3. Transaction management. The ability to start, commit, and rollback transactions when necessary. It will also support XA transactions and two-phase commits as long as you use the JDBC driver and a data source capable of XA.

To use Hibernate in our application download the latest version (currently 3.0) of the Hibernate framework. Then unzip it and add hibernate3.jar into our application's WEB-INF\lib directory. This automatically adds this JAR file to the application's build time and runtime class-path according to the J2EE standards. You also have to add dom4j.jar to WEB-INF\lib, which is also available from the Hibernate Web site. This is necessary to include the XML parser support needed for Hibernate configuration files.

Now we have to configure Hibernate on the application level. Create the hibernate.cfg.xml configuration file under "Java Source" folder in Eclipse: this way when the application is deployed it will automatically go into the application class-path under WEB-INF\classes.

The file shown in Listing 7 contains references to the following:

1. The JDBC data source reference. Note that we have to use a fully qualified reference – java:comp/env/jdbc/TestDB.
2. The request to show the generated SQL in the Java system console where System.out messages go.
3. The SQL dialect to be generated. We use the MySQL database.
4. The reference to the file that contains the mapping between domain classes and database tables.

The separate file, hibernate.mapping.xml, contains the mapping information between domain objects (entities) used in the application and the corresponding relational tables. This file should be colocated in the same directory with hibernate.cfg.xml under the Java Source.

The file in Listing 8 contains two class tags, where each of our domain classes Customer and Order is mapped to the corresponding database table, with each instance variable mapped to a database column. One attribute worth mentioning is "lazy" – we have explicitly set it to false, the reason being that when lazy is true (default), reading from the database is only done whenever a particular method is accessed. For example, it'll happen only when a getFirstName() is called, rather than pre-reading the whole set of customers right away – when the SQL query is issued. This may sometimes be beneficial when you read a large set of data and want to defer expensive database operations. In our example, we're only reading a small set of customers and don't want additional performance or database access issues to occur later, such as if the database

session is closed at that later time, we'll get an exception from Hibernate if we still try to invoke the "lazy" method.

The Hibernate configuration is complete, and we have to modify our CommandExecutor class a bit to use the framework and remove hard-coded SQL code. This class has been used as a singleton for storing data sources and getting database connections. First of all, we'll be adding the instance variable to store the Hibernate session factory. The Hibernate session factory is similar to a data source, except instead of getting database connections from it, we'll be getting Hibernate database sessions. The instance variable will look simply as follows:

```
private SessionFactory sessionFactory = null;
```

Next, we want to create an access method for this instance variable (See Listing 9). This will maintain the object state encapsulated, and will also allow using techniques of the lazy initialization (access the data only when needed):

In this method, we initialize a session factory for the first time. Hibernate's Configuration object is used to read the configuration file from the classpath and initialize the framework accordingly.

We've used the executeDatabaseCommand() method before we started using Hibernate to perform our database operations: using the DatabaseCommand interface that required the executeDatabaseOperation() method. Since we now want to use Hibernate, we're going to introduce another method to the DatabaseCommand interface and another one to the CommandExecutor singleton object – this method will execute all of our database operations using Hibernate framework (See Listing 10).

This looks very like the executeDatabaseCommand() method, except that in this case, we're using a Hibernate Session object rather than a normal JDBC connection. The next step would be simply to add the following stub method to the DatabaseCommand interface:

```
public Object executeHibernateOperation(Session session) throws SQLException;
```

Now that we've added this new method to the interface, all the classes implementing this interface will be marked with red bullets in the Eclipse workbench, because the classes implementing the interface have to implement all the methods that it requires. We have four classes implementing the database command interface:

```
public class CreateCustomer implements DatabaseCommand
public class CreateOrder implements DatabaseCommand
public class ListCustomers implements DatabaseCommand
public class ListCustomerOrders implements DatabaseCommand
```

Hence we have to add an executeHibernateOperation() implementation method to each of them. Let's take a look at the CreateCustomer class first. Its executeDatabaseOperation() method is shown in Listing 11.

This method is fairly long and coding one requires a developer to know JDBC: how to create and execute prepared statements. Also, if one were to change the database from MySQL to some other one, it might require that the developer

# Fiorano SOA™ 2006

*The Quickest Path to an SOA*

Fiorano provides the industry's first framework enabling developers to rapidly create SOA applications in well-formed J2EE code.

## Key Features:

- JCA-compliant Service-component Framework
- JMS based messaging
- BPEL support for composite components and applications
- Synchronous and Asynchronous Services in a single framework
- Support for multiple languages: Java, C, C++, C# and others

## With Fiorano you can:

- Directly deploy Service Components and BPEL processes to any standard J2EE container
- Rapidly deliver ACID transactions across front-end (JSP) and backend processes (Databases, ERPs, etc.)
- SOA enable your existing J2EE application (EJB, JCA, JMS) without reengineering or code-generation
- Easily debug complex flows spanning multiple applications and middleware
- Perform runtime service deployment, service updates and application changes



Meet us at:

**JavaOne**  
JavaOne 2006: The Java Platform for Enterprise Computing

May 16-19, 2006

Booth # 538, Moscone Center - San Francisco

**Download your copy of Fiorano today!**

<http://www.fiorano.com/downloadsoa>

# Fiorano®

Enabling Change at the speed of thought



re-write the SQL because it may differ from one database to another. With Hibernate, you just change the SQL dialect in the hibernate.cfg.xml configuration file. Our corresponding executeHibernateOperation() method is in Listing 12.

In Listing 12 we tell the session object to save our class in the database. No SQL, no JDBC knowledge, no hard-coding of column and table names. If we have to change the table or column name, we don't have to go through possibly multiple lines of code in the application. Hibernate knows how to save the object, whether this object already exists in the database or not (to do an INSERT or UPDATE operation – it performs a check in either the optimistic way (tries to do an UPDATE, and if that fails, INSERT) or pessimistic way (does a SELECT to see if the row exists, if yes, then does UPDATE, otherwise INSERT). We flush the session after the execution of the command to make sure that all the database commands are executed right away without anything left over in the framework's buffer. In a similar fashion we perform the operation for the CreateOrder class.

First, the two operations that we have dealt with (in the JDBC version), CreateCustomer and CreateOrder, are database insert operations. However, one also has to handle database queries to make the application work. For that, we have ListCustomers and ListCustomerOrders commands. Let's look at how we get the customer list.

Listing 13 contains a bunch of JDBC calls. First, create an SQL statement, execute it using hard-coded column and table names, and then go through the ResultSet of the database query and construct a Customer domain object explicitly from each row read from the table, where we also have to remember the column order or names. All those operations are error prone and may become hard to maintain whenever changes to the database tables are required. This is where Hibernate comes to the rescue. It introduces a whole new language called the Hibernate Query Language (HQL), where one doesn't need to query database tables, but rather objects. Our executeHibernateOperation() method will look as in Listing 14.

Once again this method looks pretty similar to the old one, however, there are some significant differences. Here we are

using a bunch of Hibernate objects. The first one is a Query class. It lets one create and execute database queries using either HQL (through the createQuery method) or regular SQL (through the createSQLQuery method). Let's look at the HQL we are using here:

```
from customer in class domain.Customer
```

Basically we're selecting all the customers identified by the variable customer from the domain.Customer class. Obtaining the iterator of the query lets us put "customers" in any collection. In our case it's ArrayList<Customer>.

A very similar method can be written for the ListCustomerOrders class, but HQL is a tiny bit more complex as you can see in Listing 15.

In this case we use the where-clause in our query. Note that in this where-clause we can use the instance variable of the Order class (custId) to query by. The syntax is similar to Java's dot notation.

Finally we have to update our Struts action classes to invoke the executeHibernateOperation() method instead of the executeDatabaseOperationMethod(). This can be done easily using Eclipse editors.

## Exporting an Eclipse Project into WAR

Using WTP tools we can easily export our project into the WAR file to be deployed under Tomcat. Just select the DBTestStrutsWeb project, then select Export from the File menu. When prompted, select the WAR file, specify the file name, and you'll have the WAR file ready to be deployed in Tomcat.

## Conclusion

The goal of this article was to show you how to integrate Struts and Hibernate support into a simple Web application developed using Eclipse and WTP tooling. These frameworks help to improve application maintainability and code reusability, as well as code clarity. ☺

### Listing 1: Struts Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD
Struts Configuration 1.1//EN" "http://jakarta.apache.org/struts/dtds/
struts-config_1_1.dtd">

<struts-config>
  <!-- Data Sources -->
  <data-sources>
  </data-sources>

  <!-- Form Beans -->
  <form-beans>
  </form-beans>

  <!-- Global Exceptions -->
  <global-exceptions>
  </global-exceptions>

  <!-- Global Forwards -->
  <global-forwards>
  </global-forwards>

  <!-- Action Mappings -->
  <action-mappings>
  </action-mappings>
</struts-config>
```

### Listing 2: Enabling Struts Support in web.xml

```
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-
class>
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <init-param>
    <param-name>debug</param-name>
    <param-value>2</param-value>
  </init-param>
  <init-param>
    <param-name>detail</param-name>
    <param-value>2</param-value>
  </init-param>
  <init-param>
    <param-name>validate</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

# Visualize works of software art

Draw on UModel® 2006, and picture  
better programs based on UML™.

## New in Version 2006:

- UML 2.1 support
- C# code generation, reverse engineering, and round tripping
- Sequence diagrams
- Interoperability via XMI 2.1

Altova UModel 2006, the intriguing new entrant in the software design space, is defying convention and earning acclaim as the simple, cost-effective way to draw on UML. Use it to interpret or create software architectures. Decode Java or C# programs into clear, accurate UML2 diagrams, or outline applications and generate code from your plans. With customizable templates, built-in entry helpers, and many other artful features, UModel makes visual software design practical for programmers and project managers. Take the mystery out of UML!

Download UModel® 2006  
today: [www.altova.com](http://www.altova.com)

Visit Altova  
at JavaOne  
booth # 626

UML is a trademark or registered trademark  
of the Object Management Group, Inc. in the  
United States and other countries.

# Rich Internet Applications with Adobe Flex 2 and Java

(((Flash talks to POJO

by Yakov Fain, Victor Rasputnis,  
and Anatole Tartakovsky

A typical Java developer knows that when you need to develop a GUI for a Java application, Swing is the tool. Eclipse SWT also has a number of followers, but the majority of people use Java Swing. For the past 10 years, it was a given that Swing development wouldn't be easy; you have to master working with the event-dispatch thread, GridBagLayout, and the like. Recently, the NetBeans team created a nice GUI designer called Matisse, which was also ported to MyEclipse. Prior to Matisse, JBuilder had the best Swing designer, but it was too expensive. Now a good designer comes with NetBeans for free.

Why even consider Flex for developing Rich Internet Applications (RIA)? First, we'll give the short answer. Just look at the code in Listing 1. This code compiles and runs in the Flash player and produces the output shown in Figure 1. Yes, it's a tree control with several nodes that know how to expand, collapse, and highlight the user's selection. Imagine the amount of Java code you'd need to write to achieve the same functionality.

The code is nice and clean; the GUI looks rich and appealing. The Flex compiler automatically converts the MXML code in Listing 1 into an object-oriented language called ActionScript, and then compiles it into an SWF file, the format that the Flash player understands.

To add the business processing to this example, we'd need to write event handlers in the ActionScript 3.0 language, which is very similar to Java, but we wouldn't have to worry about routing all events to the event-dispatch queue. Below are some of the other reasons that the Flex/Flash combination is a very promising technology for RIA development:

- The Flash player, a powerful virtual machine with a high-performing byte code/JIT compiler and rich UI API, is available on most of the platforms.
- The size of the VM is small.
- It easily integrates with Web browsers.
- Flash applications can run outside of the Web browser.
- Flex offers component-based programming, which eliminates most of the low-level coding.
- It offers simple integration with all kinds of media (video and audio). Java Swing is behind in this field.
- It has a quick adoption rate as all other Adobe products.
- Most important, Flex 2 easily integrates with Java on the server side.

Later in this article we'll develop a Stock Portfolio Application with Flex, and then we'll integrate it with POJOs.

Flex client applications are compiled SWF files that can be delivered to the client and run by the Flash player, which is installed as a plug-in to your browser. On the client side, Flex consists of a Flash player, a framework of predefined components, a couple of command-line compilers, and an Eclipse-based Flex Builder IDE. On the server side, Flex is a Web application that includes Flex Data Services and a Flex Charting component and can be deployed in any JEE server.

To see Flex in action, let's write the functional specification, then develop and deploy a sample application.

## Designing the Stock Portfolio Application

We'll create a Web application that will receive and display a feed containing security prices and the latest news as in Figures 2 and 3.

The top portion of the screen must be populated by the stocks included in the user's portfolio. For simplicity, store the user's portfolio in the XML file as in Listing 2.

When the user clicks on a row with a particular stock (i.e., ADBE as in Figure 2), populate the lower data grid with the headlines related to the selected stock. The news should be coming from <http://finance.yahoo.com/rss/headline>. The column link has to contain the URLs of the news, and when the user clicks on the link, a new browser's window should pop up displaying the selected news article.

The top of the screen should contain the toggle buttons Show Grid/Show Chart. When the Show Grid option is selected, the user will see the screen as in Figure 2, and



**Yakov Fain** is a principal consultant of Farata Systems. He's responsible for the enterprise architecture and emerging technologies. Yakov has authored several Java books, dozens of technical articles, and his blog is very popular. Sun Microsystems has nominated and awarded Yakov with the title Java Champion. He leads the Princeton Java Users Group. Yakov holds a BS and MS in applied math.

[yfain@faratasystems.com](mailto:yfain@faratasystems.com)

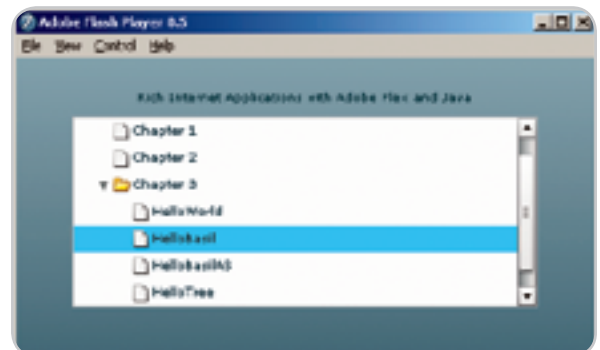


Figure 1 Running HelloTree in Flash player



# Bridge the Gap Between Windows and Java

Give your users the freedom to use Microsoft Windows based reporting, analysis, and development applications against your Java data source

OpenAccess™ SDK lets you quickly implement a custom ODBC driver with full SQL support for any data source or application.

## Key OpenAccess™ SDK Features & Benefits

- Write the ODBC driver with SQL support in just weeks  
- 99% of the required code is contained in the SDK.
- Use Java to implement the data source specific code.
- Guaranteed to work with ODBC compliant applications  
- hundreds of organizations have used OpenAccess™ to set their data free.
- Fully Unicode enabled for globalization.
- When required easily add JDBC, OLE DB, or .NET drivers.
- Designed from ground up for reliability, scalability, and portability.

OpenAccess can also be used to add ODBC support on top of your JDBC driver and to add a security layer in between a client application that uses ODBC or JDBC and the data source. Please call us or visit our web site to learn how different problems have been solved using OpenAccess.

To learn more about OpenAccess™ SDK for Java or to try it out, go to [www.OpenAccessSoftware.com/JDJ](http://www.OpenAccessSoftware.com/JDJ) or call 1.888.805.ODBC

Crystal Reports  
Access  
Excel  
.NET



ODS  
BQL  
CL

00110100  
01000100  
Java  
10100010  
00010101

**OpenAccess**  
SOFTWARE

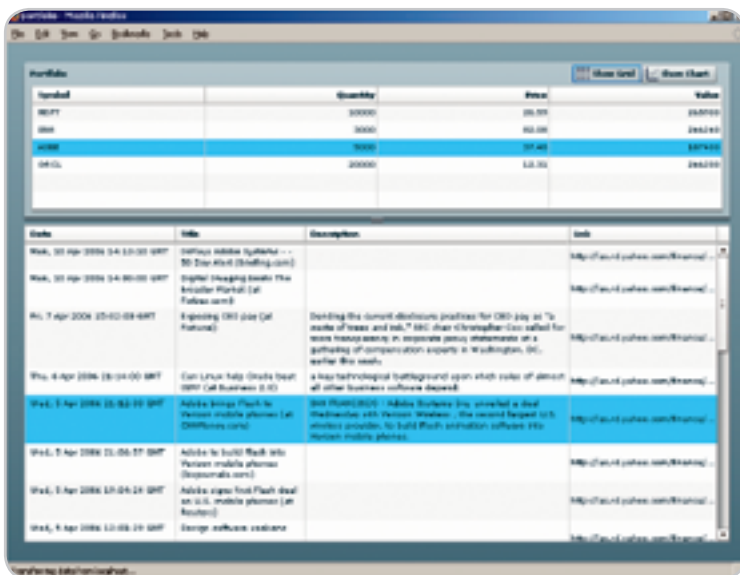


Figure 2 Stock Portfolio Screen: Show Grid view

when Show Chart is selected, the top data grid has to be replaced with the pie chart (see Figure 3), preserving the same functionality (clicking on the pie slice repopulates the news headlines). The market data has to be refreshed on the screen every second or so.

Since we are limited by the size of this article, for the server side we'll just write a simple POJO that will generate random numbers and push them back to the Flash client. However, in the next article we'll add the data feed from an external Java application via JMS.

Sounds like an ambitious task for a short magazine article, doesn't it? This article is not a tutorial on Flex MXML or ActionScript, but as a Java developer, you should be able to understand how the code is put together with minimal explanations. After deploying and running this application (the source code is located at <http://java.sys-con.com/read/210991.htm>) you can study Flex more formally. So let's roll up our sleeves...

### Installing Flex Builder and Flex Data Services

At the time of this writing, Flex 2 is still in beta (<http://labs.macromedia.com/flexproductline/>) and its production release is expected this summer. The burning question is, "Is it free?" The answer is, "It depends." There is a free lunch (or rather a complimentary appetizer from the chef), but it'll just whet your appetite, and pretty soon your hand will start slowly reaching for the wallet. Flex Data Services (FDS) that provide seamless integration of Flex into enterprise data and messaging have deployment fees. For high-performance enterprise applications supporting client authentication, messaging, and the like, you may want to purchase the FDS license. For smaller applications, you might use integration via HTTPService, Web Services, and Remote Java invocation using standard and open source technologies.

The free part is Flex command-line compilers and the Flex framework that includes all these libraries of rich components. This means you can use any plain text editor for writing Flex code, compile it into the SWF file, and run it in a free Flash 8.5 player. As of today though, Adobe says that the Eclipse-based

Flex Builder IDE, which makes development a lot more productive, will not be free. Flex Charting components that offer client-side interactive charts are not free either.

The Flex Builder IDE comes in two flavors: a standalone version and the Eclipse plug-in. You'll have to choose one of these versions during the installation process. Let's install Eclipse 3.1 first from <http://www.eclipse.org/downloads/> (unless you have it already), and then the plug-in version of Flex Builder. During the installation, you'll have to specify the directory where your Eclipse resides (the one that has a sub-directory called plug-ins). After Flex Builder is installed, start the Eclipse IDE, select the menus Window | Open Perspective | Other and add the Flex Development perspective (the Flex Debugging perspective will be added automatically to the Eclipse IDE). The Flex Builder comes with great Help content, which in the plug-in version is hidden under the menu Help | Help Contents | Adobe Help.

To run client-only Flex applications, we need to install one more program: Flash Player 8.5. However, for our stock portfolio application, we'll need to install and deploy FDS, which is a Web application that can be deployed in any JEE compliant container. Since FDS is a Web application, you can deploy it in the application server of your choice. First, we'll download and install Apache Tomcat 5.5 from <http://tomcat.apache.org/>. By default, on Windows machines Tomcat is installed in "C:\Program Files\Apache Software Foundation\Tomcat 5.5" and runs on port 8080. At the end of the installation, the Apache Tomcat server will be started as a Windows service. Now download from and run the FDS executable installer (default directory ID C:\fds2), picking the optional Flex Data Services J2EE Application. After the installation is finished, unzip the content of the flex.war into Tomcat's webapps\ROOT directory and restart the Apache Tomcat service. Enter the URL <http://127.0.0.1:8080/> in your browser, and you should see the FDS Welcome screen. Flex Data Services are deployed now under Tomcat.

### Developing Stock Portfolio Application with Flex

#### What Runs Where

First, let's see which components will run on the client and what belongs to the Web server. The code shown in Listings 3–6 will be compiled into portfolio.swf, which will run on the client either independently in the Flash 8.5 player, or with the help of portfolio.html (auto-generated by Flex Builder) in the Web browser with the Flash plug-in. Tomcat is our server, which will host compiled Java classes shown in Listings 7 and 8. We'll also make small additions in the server configuration files (see Listings 9 and 10) to specify where to find the Java classes and to allow access to the external Yahoo! service. While Flex provides different ways of client/server communication (RemoteObjects, HTTPService, WebService), in this article our client will use the RPC by means of the <mx:RemoteObject> component that will find the matching Java classes located and configured on the server as described at the end of this article.

### Developing the GUI Part

Open the Flex perspective in Eclipse and create a new project (e.g., Portfolio\_RCP) containing files from Listings 3–7, and then compile it into one file called portfolio.swf. To simplify deployment, set the project's output directory to Tomcat's \ROOT\portfolio.



**Victor Rasputnis** is a managing principal of Farata Systems. He provides architectural design, implementation management, and mentoring to companies migrating to XML Internet technologies. He holds a PhD in computer science from the Moscow Institute of Robotics.

[vrasputnis@faratasystems.com](mailto:vrasputnis@faratasystems.com)



Our main file portfolio.mxml (see Listing 3) divides the screen with the vertical Flex box-type layout (<mx:VDivided-Box>) with an adjustable divider between its children (think of Swing's split panes), and it includes two separate code fragments. The bottom child contains a data grid programmed in <FinancialNews> (see Listing 6). The top child contains <PortfolioView> (see Listing 4).

Java developers know that some Swing components store the data and GUI parts separately, for example, the JTable gets its feed from a data model class that can store its data in a Java collection. On a similar note, some Flex controls (e.g., <mx:DataGrid>) can also use ActionScript collection classes as data providers. Our data model is the file portfolio.xml depicted in Listing 2.

This code represents a typical master-detail relationship, where a change in the selected security in the master component <PortfolioView> repopulates the detail component <FinancialNews>.

## Getting the Price Quotes

Let's go over some interesting programming techniques that we've applied while coding the top panel, which contains several Flex framework components wrapped into a Canvas container (see Listing 4). After learning that the user's portfolio is represented by portfolio.xml, you may expect at least some number of lines performing XML parsing.

What do you think of this line:

```
<mx:XML format="e4x" id="portfolioModel" source="portfolio.xml" />
```

The XML parsing is complete! After this line you can traverse the entire XML document using a simple dot notation, e.g., portfolioModel.security. This magic is possible because Flex supports the e4x format that was introduced by W3C as a simple alternative for reading and processing XML documents.

Another interesting feature to comment is data binding, which is a way to link the data in one object to another. In particular, it's a convenient way to move the data between the GUI and non-visual components. Data binding can be specified using the curly braces syntax. The following lines from Listing 4 provide an immediate refresh of the top data grid and chart as soon as the data in the underlying security changes.

```
<mx:DataGrid id="portfolioGrid" width="100%" height="100%"
```

```
dataProvider="{portfolioModel.security}"
```

```
<mx:PieChart id="portfolioPie" dataProvider="{portfolioModel.security}"
```

The variable portfolioModel.security represents the data source, and the destination will change as soon as any property of the source changes. Note that we have one data source mapped to two destinations (the data grid and pie chart), so we don't need to do any additional programming to refresh the data grid and the chart when the incoming market data changes the properties of the security.

The tag <mx:RemoteObject destination="Portfolio" id="freshQuotes" > declares the connection of our client to a remote object that resides on the server and goes by the name "Portfolio" (see Listing 10).



Figure 3 Stock Portfolio Screen: Show Chart view

Listing 4 has the ActionScript code embedded into mxml in the tag <mx:Script>. It has a public variable marked with a [Bindable] metadata:

```
[Bindable] public var selectedSecurity:String;
```

This variable will automatically send notifications as soon as its value changes to all registered listeners. This may happen as a result of the change event in the data grid or if the user clicks on the slice of the pie. Selecting a different stock symbol (security) will trigger the repopulation of the news grid. No additional programming is required. Wait a minute, we've never programmed any event listeners for the variable selectedSecurity! That's correct, but when Flex compiles Portfolio.mxml (see Listing 3), it'll notice the bindable public variable pv.selectedSecurity, so it'll not only generate a registered required listener, but it will also call the function set security in the FinancialNews panel (see Listing 6).

The show begins as soon as the painting of the top portion of the screen (<mx:Canvas>) completes. Its creationComplete event handler calls the ActionScript function startQuotes(), which connects to the remote POJO and calls its method getQuotes() every second. The tag <mx:RemoteObject> also contains the element declaring how the method getQuotes() should be invoked. As soon as the result of this method call arrives, it's being passed to the function applyQuotes(). All Flex remote calls are asynchronous and require an ActionScript method that will process the result of the call. In our example, applyQuotes() is such a method. The concurrency attribute defines how the remote object should process multiple requests. For example, the user requested quotes for one security, but it takes a bit longer than usual, and she sent a quote request for another security. The attribute concurrency="last" means that older responses should be ignored.

Another interesting line illustrating ActionScript capabilities is e4x object access in Listing 4:

```
var row:* = portfolioModel.security.(symbol==quote.symbol);
```



Anatole Tartakovsky is a managing principal of Farata Systems. He's responsible for the creation of frameworks and reusable components. Anatole has authored a number of books and articles on AJAX, XML, the Internet, and client/server technologies. He holds an MS in mathematics.

atartakovsky@faratasystems.com



Flex is a great platform for any XML-related processing. It'll automatically handle `Symbol==quote.symbol` as XPath expressions behind the scenes, providing a code-free approach for data navigation.

Just to tease you, we've decided to keep commented lines `<mx:Consumer>` and `consumer.subscribe()` in the code to give you an idea what has to be added to replace the AJAX-style price quote polling performed by `setInterval()` with a server push implemented by publish/subscribe messaging. We'll cover this topic in our next article.

The top toggle buttons Show Grid/Show Chart (see Figure 2) use images, and it's a good idea to embed these resources right into the SWF file. This way the browser can download the entire client in one HTTP request (but the size of the SWF file becomes larger). For the same reason multi-file Java applets are packaged into a single JAR. There are two lines in Listing 4 that embed the images and assign them to reference variables. PortfolioView displays them as follows:

```
<mx:VBox label="Show Grid" icon="iconGrid" ...>
    <mx:HBox label="Show Chart" icon="iconChart" ...>
```

Listing 5 contains the ActionScript code that's used on the client just to help Flex perform Java introspection more efficiently; we've declared a class with all properties that exist in its Java peer on the server (see Listing 8).

## Getting the Financial News

The bottom data grid (see Listing 6) displays the news headlines supplied by the Web site. This time we're not using `<mx:RemoteObject>` but another flavor of RPC called `<mx:HTTPService>` that points at <http://finance.yahoo.com/rss/headline>, which is known to clients as YahooFinancialNews (see Listing 9). Again, the HTTPService call works asynchronously: as soon as security setter is called (set security), it sends the request `newsFeed.send({s: value})` to the server. In this line, "s" is the name of the parameter, and the value should contain the selected security. For example, if the user selects MSFT, the server destination YahooFinancialNews will receive the following URL: <http://finance.yahoo.com/rss/headline?s=MSFT>. Just try to enter this URL manually in your Web browser to see the XML our Flash client will receive as a result of this call.

Received XML is used as a source property of the `<mx:XMLListCollection>`. The data grid's columns are mapped to the nodes of this data provider. The collection and then GUI are refreshed upon arrival of the result (a property of `newsFeed`). You don't have to use a collection as a middleman between the data and the GUI component. However, collections may become handy if you'd like to perform some additional data massaging before presenting it to the user, e.g., filtering or search using `e4x`.

The following line puts the result of the HTTPService request (`newsFeed.result`) into the source property of an `XMLListCollection`:

```
<mx:XMLListCollection id="newsList"
    source="{newsFeed.result.channel.item}" />
```

If you look again at the structure of the XML returned by the URL <http://finance.yahoo.com/rss/headline?s=MSFT>, you'll notice that the channel is the root element there and each headline is represented by the element called item.

The last column in the A composite element is called `<mx:LinkButton>`. It's a button that looks like a hyperlink. We've carefully wrapped it into a container `<mx:Component>`, which in turn sits inside `<mx:ItemRenderer>` (imagine a Swing cell renderer on steroids). Click on this link to navigate to this URL, which will be opened in the popup window. The argument of the `URLRequest` constructor is `data.link`; data is a reference to the current row in the data, and link is the name of the column.

Now open the Java perspective in Eclipse and create a new project (e.g., `Portfolio_RCP_RO`), which will contain our Java classes. The POJOs that live in the Tomcat server are very simple. The `StockQuoteDTO.java` (see Listing 7) contains the last price of a particular stock. The class `Portfolio.java` (see Listing 8) is a simple, random number generator simulating market-like real-time price changes for several hard-coded securities. Compiled Java classes should go under Tomcat's directory `WEB-INF\classes`.

## Configuring the Server-Side Destination and Proxy

For security reasons, Flash clients can only access the servers they came from, unless there is an agreement between our server and external service providers, which agreed to deal with our server and are declared in a `crossdomain.xml` file. However, our portfolio SWF was not loaded from `finance.yahoo.com`, and we are not allowed to install `crossdomain.xml` on the Yahoo! Servers. We'll use another technique called Flex proxy. When the user clicks on the News link in the data grid, the portfolio client will connect to our FDS application deployed under Tomcat, which will proxy our communication with Yahoo!. To configure the Flex proxy service, see the following section to the `flex-proxy-service.xml` located in the Tomcat's `WEB-INF\flex` directory (see Listing 9).

Now FDS will contact <http://finance.yahoo.com>, get the news for a specified symbol, and return it back to the Flash client.

As for POJO on the server side, Flex provides configuration files that allow you to hide the exact service providers details (e.g., actual Java class names) by specifying so-called destinations. The following section from Listing 10 has to be added to the `flex-remoting-service.xml` file.

Clients won't know that the actual name of our POJO is `com.theriabook.ro.Portfolio`, but they'll be able to refer to it by the nickname `Portfolio`.

Now start your Tomcat service and run the Flex application (`portfolio.mxml`) in the Flex Builder's project `Portfolio_RCP`. You should be able to see the screens as in Figures 2 and 3, the "market data feed" should modify the prices and you'll be able to read the latest Yahoo! news on your stocks.

## Conclusion

In this short article we managed to develop an application with not so trivial functionality, but the amount of code we had to write was minimal. Unfortunately, the amount of explanation we could provide was minimal as well. If you'd like to start learning Flex 2, you can find lots of quality introductory materials and product documentation at [http://labs.macromedia.com/wiki/index.php/Main\\_Page](http://labs.macromedia.com/wiki/index.php/Main_Page). Our upcoming book *Rich Internet Applications with Adobe Flex and Java* (<http://www.theriabook.com>) will show how to face-lift your enterprise Java applications using advanced and not-so-obvious Flex techniques. ☺

**Listing 1: HelloTree.mxml**

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" xmlns=""
layout="vertical">

<mx:Label text="Rich Internet Applications with Adobe Flex and Java"/>
<mx:Tree id="tree1" labelField="@label" showRoot="false" width="400">

    <mx:dataProvider>
        <mx:XML format="e4x">
            <folder label="TOC">
                <folder label="Chapter 1"/>
                <folder label="Chapter 2"/>
                <folder label="Chapter 3">
                    <Sample label="HelloWorld" />
                    <Sample label="HelloBasil" />
                    <Sample label="HelloBasilAS" />
                    <Sample label="HelloTree" />
                </folder>
                <folder label="Chapter 4" />
            </folder>
        </mx:XML>
    </mx:dataProvider>
</mx:Tree>

</mx:Application>

```

**Listing 2: A fragment of portfolio.xml**

```

<portfolio>
    <security>
        <Symbol>MSFT</Symbol>
        <Quantity>10000</Quantity>
        <Price>20.56</Price>
        <Value>1</Value>
    </security>
    <security>
        <Symbol>IBM</Symbol>
        <Quantity>3000</Quantity>

        <Price>80.21</Price>
        <Value>1</Value>
    </security>
    ...
</portfolio>

```

**Listing 3: Portfolio.mxml**

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
xmlns="" layout="vertical">
    <mx:VDividedBox width="100%" height="100%">
        <PortfolioView id="pv"/>
        <FinancialNews id="fn" security="{pv.selectedSecurity}"/>
    </mx:VDividedBox>
</mx:Application>

```

**Listing 4: The top panel's content: PortfolioView.mxml**

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml" xmlns=""
width="100%" height="100%" creationComplete="startQuotes();" >
<mx:XML format="e4x" id="portfolioModel" source="portfolio.xml" />
<mx:Panel width="100%" height="100%" title="Portfolio">
    <mx:ViewStack id="vs" width="100%" height="100%">
        <mx:VBox label="Show Grid" icon="iconGrid" >
            <mx:DataGrid id="portfolioGrid" width="100%" height="100%"

            dataProvider="{portfolioModel.security}"
            change="selectedSecurity = portfolioGrid.selectedItem.Symbol;"/>

```

```

        <mx:columns><mx:Array>
            <mx:DataGridColumn dataField="Symbol"/>
            <mx:DataGridColumn dataField="Quantity"
                textAlign="right"/>
            <mx:DataGridColumn dataField="Price"
                textAlign="right"/>
            <mx:DataGridColumn dataField="Value"
                textAlign="right"/>
        </mx:Array></mx:columns>
    </mx:DataGrid>

    </mx:VBox>
    <mx:HBox label="Show Chart" icon="iconChart" horizontalAlign="center"

    verticalAlign="middle">
        <mx:PieChart id="portfolioPie" dataProvider="{portfolioModel.secu-
            rity}" showDataTips="true"
            itemClick="selectedSecurity=event.hitData.item.Symbol"
            height="90%">
            <mx:series>
                <mx:Array>
                    <mx:PieSeries labelPosition="callout"
                        field="Value" labelFunction="showPosition"

                        nameField="Symbol" explodeRadius="2"/>
                </mx:Array>
            </mx:series>
        </mx:PieChart>
        <mx:Legend verticalAlign="middle" dataProvider="{portfolioPie}"
            label="{data.Symbol}"/>
    </mx:HBox>
</mx:ViewStack>
</mx:Panel>
<mx:HBox horizontalAlign="right" width="98%" >
    <mx:ToggleButtonBar dataProvider="{vs}" paddingTop="4" />
</mx:HBox>
<mx:RemoteObject destination="Portfolio" id="freshQuotes" >
    <mx:method name="getQuotes" concurrency="last"
        result="applyQuotes(event.result as Array)"/>
</mx:RemoteObject>

<!-- mx:Consumer id="consumer" destination="chat-topic-jms"
    message="applyQuotes(event.message.body.quotes)" / -->

<mx:Script><![CDATA[
    import mx.controls.Alert;
    [Bindable] public var selectedSecurity:String;
    private function showPosition(data:Object, field:String, index:
    Number, percentValue:Number):String
    {
        return data.Symbol + "\n" + "Shares:" + data.Quantity + "\n" +
        "Price:" + data.Price + "\n" +

        "Value:" + data.Value ;
    }
    [Embed(source="images/icon_chart.png")] public var iconChart : Class;
    [Embed(source="images/icon_grid.png")] public var iconGrid : Class;

    import com.theriabook.jms.dto.StockQuoteDTO;
    private function applyQuotes(quotes: Array):void {
        for (var i:int=0; i<quotes.length; i++) {
            quote = StockQuoteDTO(quotes[i]);
        }
    }
    //casting
    var row:* = portfolioModel.

```

```

security.(Symbol==quote.symbol);
        row.Price = Math.round(100*quote.last)/100;
        row.Value = Math.round(row.Price * row.
        Quantity);
    }
}

internal var quote:StockQuoteDTO = null;
private function startQuotes():void{
    setInterval(function () { freshQuotes.
    getQuotes(); }, 1000);

    //consumer.subscribe();
}

}]></mx:Script>
</mx:Canvas>

```

#### Listing 5: The ActionScript on the client: StockQuoteDTO.as

```

package com.theriabook.jms.dto {

[Managed]
[RemoteClass(alias="com.theriabook.jms.dto.StockQuoteDTO")]
public dynamic class StockQuoteDTO
{
    public var symbol:String;
    public var last:Number;
}
}

```

#### Listing 6: The bottom panel's content: FinancialNews.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml" xmlns:*="*" width="100%"
height="100%">

    <mx:DataGrid id="newsGrid" width="100%" height="100%"
        dataProvider="{newsList}" variableRowHeight="true">

        <mx:columns>
            <mx:Array>
                <mx:DataGridColumn headerText="Date"
                    dataField="pubDate" width="200"/>
                <mx:DataGridColumn headerText="Title"
                    dataField="title" wordWrap="true" />

                <mx:DataGridColumn headerText="Description"

                    dataField="description" wordWrap="true" />

                <mx:DataGridColumn headerText="Link"
                    width="130">
                    <mx:itemRenderer>
                        <mx:Component>
                            <mx:LinkButton label="{data.
                                link}"

                                click="navigateToURL(new URLRequest(data.link), '_blank')"/>
                        </mx:Component>
                    </mx:itemRenderer>
                </mx:DataGridColumn>
            </mx:Array>
        </mx:columns>
    </mx>DataGrid>

    <mx:XMLListCollection id="newsList"
        source="{newsFeed.result.channel.item}" />
    <mx:HTTPService id="newsFeed" useProxy="true"

        destination="YahooFinancialNews" concurrency="last"
        resultFormat="e4x" >

```

```

</mx:HTTPService>
<mx:Script>
    <![CDATA[
        public function set security(value:
        String):void {
            newsFeed.send({s:value});
        }
    ]]>
</mx:Script>
</mx:Canvas>

```

#### Listing 7: The Java counterpart: StockQuoteDTO.java

```

package com.theriabook.jms.dto;
import java.io.Serializable;
public class StockQuoteDTO implements Serializable {
    private static final long serialVersionUID =
    4672447577075475117L;
    public String symbol;
    public double last;
    public StockQuoteDTO(String sym, double newPrice){
        symbol = sym;
        last = newPrice;
    }
}

```

#### Listing 8: A simple stock quotes generator: Portfolio.java

```

package com.theriabook.ro;
import java.util.Random;
import com.theriabook.jms.dto.StockQuoteDTO;

public class Portfolio {

    static Random random = new Random();
    static StockQuoteDTO[] quotes = {
        new StockQuoteDTO("IBM", 82.0),
        new StockQuoteDTO("MSFT", 27.0),
        new StockQuoteDTO("ADBE", 38.0),
        new StockQuoteDTO("ORCL", 13.0));

    double volatility=.05;
    public StockQuoteDTO[] getQuotes() {
        for (int i = 0; i < quotes.length;i++){
            quotes[i].last += random.nextGaussian()*
            volatility;
        }
        return quotes;
    }
}

```

#### Listing 9: Configuring proxy to allow access to an external server

```

<destination id="YahooFinancialNews">
    <properties>
        <url>http://finance.yahoo.com/rss/headline</url>
    </properties>
</destination>

```

#### Listing 10: Configuring Flex remoting service

```

<destination id="Portfolio">
    <properties>
        <source>com.theriabook.ro.Portfolio</source>
    </properties>
</destination>

```



# Lacking visibility in your development process?



## **Enerjy CQ2 delivers complete visibility for a quality-driven development process**

Enerjy CQ2 integrates with your version control system and existing tool set. It runs parallel with existing project build systems to collect, store and graphically display code quality metrics.

### **Enerjy CQ2 helps to produce the highest quality applications by measuring**

- Adherence to coding standards
- Unit test results
- Code coverage metrics
- Developer trends and activity

## **Enerjy CQ2 presents metrics in a centralized Web application**

- Provides instant access to data reports and analysis
- Graphically displays metrics and trends on a per project, per team and per developer basis
- Leverages version control system to identify code ownership

---

**ENERJY<sup>®</sup>**  
Software Integrity

To learn more about Enerjy CQ2 visit  
[www.enerjy.com/visible](http://www.enerjy.com/visible)

by Laurence Moroney

# J2EE/.NET Interoperability

## *A look at managing complex interoperating systems*

**D**ue to the benefits of each, J2EE and .NET have penetrated most markets and companies to the point where 95% of medium and large-scale enterprises support both .NET and J2EE, and 30% or more of new application development will include both by 2009, according to a study published by Gartner. Data centers of these companies rarely work in "silo" mode where J2EE and .NET work independently and don't need to interoperate with each other, but instead form a mesh of applications in what is termed a "mixed-mode" deployment. These deployments have driven the emergence of standards such as Web Services to ease their integration.

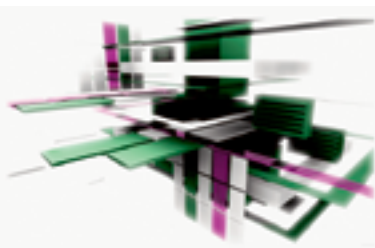
To ensure that applications can run across such diverse platforms and have acceptable availability, performance, scalability, and security characteristics is quite a challenge. Several technologies are typically used to enable interoperability between .NET- and J2EE-based systems. However, each solution introduces manageability issues. Platform unification, on the other hand, attacks interoperability problems at their root by removing the need for complex interoperability through a unified runtime application stack.

### Why Do We Have Mixed-Mode Systems?

Complex mixed-mode data centers arise for a variety of reasons. They are rarely deliberately mixed-mode by design. Instead, they become complex over time with the constant addition of new products meeting new technical, product, and business requirements. For example, when a company is considering a merger or an acquisition, the primary driver is business alignment. Typically the underlying technology used to run the data centers and the products of each business party is not a major factor, and only after the business transaction is complete are the relevant IT organizations handed the task of making hugely different technology bases somehow integrate to form a

coherent whole. It's not uncommon for a company that uses J2EE end-to-end to find itself acquiring or acquired by one that uses .NET through and through, and now they have to make their systems work together.

Sometimes changes in management or strategy are the reason a company ends up with a mixed-mode system. As technology evolves, a company that is vested in a particular strategy may have many reasons for wanting to change it. For example, companies that don't like the complexity of J2EE development may be seduced by the relative simplicity of .NET and move new product



development to the Microsoft platform. Conversely, companies concerned about the security holes found in IIS may prefer to move to J2EE. After this decision, their new product development will still have to interoperate with the old products built on the "other" technology.

Also, companies that use one technology coherently on the strategic corporate level, but allow other technologies on the department level can experience great success with some department-level applications. Should they want to extend these to the rest of the enterprise, they will require an interoperability or migration strategy.

Others are service-driven when implementing different platforms. Many companies want to simplify their intra-company interoperability, or their interface to their customers by providing Web Services. Integrating with other business partners or customers who use different technology bases will require a cross-technology interoperability solution.

And naturally, cutting costs is always a major business driver. As new technology bases arrive on the scene, and can cut development costs relative to older ones, companies will jump aboard to take advantage of the cost savings. However, in many cases new products will still have to interoperate with older ones built on different platforms.

### Interoperability Strategies and Solutions

The interoperability problem is a common one, and as such there are many solutions available from software vendors to ease and solve them. When considering a cross-platform solution, one must consider fault management, configuration management, accounting management, performance management and security management.

### Web Services

Web Services has emerged as the ipso facto standard in connecting disparate systems due to their abstract nature: they use XML throughout from description to discovery, and their underlying technical implementation is entirely abstracted as a result. Hence they would appear to be the ideal solution for interoperability. If your application can be implemented on a platform that supports Web Services, or at least "wrapped" in one, then this provides an excellent, though limited solution. Limited because Web Services, being stateless by nature, and not message-driven can't be used in all circumstances. Additionally, technology implementations by different vendors lead to commonly reported interoperability problems when moving across the J2EE/.NET divide.

From a management point-of-view, the following observations are made with regard to Web Services:

- **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

Lawrence Moroney is the

director of Technology

Evangelism at Mainsoft

and the author of several

books on .NET and Web

services, as well as several

dozen articles that span

the technology realm. He

believes strongly that good

interoperability is the key to

open standards and open

source working efficiently and

effectively, and that this interop

should begin at the human

level. This makes him a keen

enthusiast of the Visual MainWin

for J2EE product, because it

allows developers to work

together more effectively to

deliver systems that operate

on diverse platforms. Prior to

working for Mainsoft his career

spanned many different domains,

including architecture for

financial services systems,

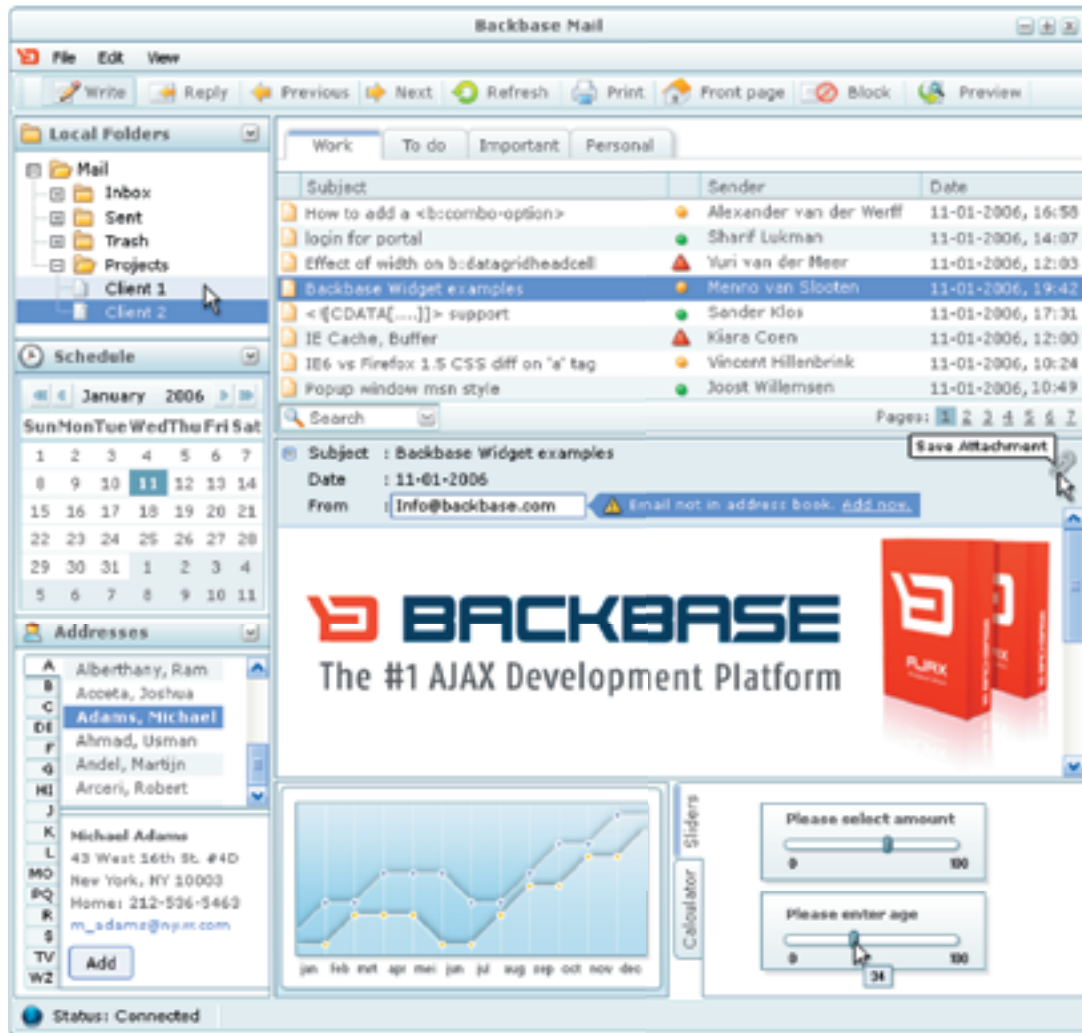
airports, casinos, and professional

sports.

lmoroney@mainsoft.com



# Create Rich AJAX User Interfaces, Quickly



Eclipse Plug-in  
for Fast Development



Server-side Extensions  
for Java Server Faces



Powerful and Scalable  
AJAX Components

Download free trial today at [www.backbase.com/jsf](http://www.backbase.com/jsf)

i: [www.backbase.com](http://www.backbase.com) | t: (866) 800-8996 | e: [sales-us@backbase.com](mailto:sales-us@backbase.com)



the case of management and fault management, WS-Management and Management of Web Services (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 addresses 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 their need 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.
- **Security Management:** There are many emerging standards that help Web Services be secure: WS-Management defines a schema format that allows message content to be encrypted and signed for security, non-repudiation, and identity. Web Services security still has to face the challenge of single sign-on where processes that run across diverse platforms have to authenticate themselves on each platform effectively. Great strides are being made in this area, but there's still a long way to go.

### 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 front-end presentation tiers are developed using ASP.NET and the back-end business logic and data management tiers are implemented on J2EE, 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 does suffer from challenges insofar as management is concerned.

Consider 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.

Where configuration management is concerned, 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 carefully deployed and managed on both sides of the system, and in many cases may need to be regenerated upon small changes to the system.

Generally bridging solutions don't integrate with the underlying management APIs, and don't offer a facility for unified accounting.

And, performance is sometimes compromised. Bridging solutions bring an overhead to the performance of the system that is significant, and in some cases major. The runtime data marshaling can reduce overall performance.

Plus, security issues arise because security context management handling can vary from implementation to implementation, but when using a bridging solution, a new channel of communication is introduced that may or may not allow for a compatible security context with the rest of your J2EE or .NET architecture.

### EAI Applications

There are many Enterprise Application Integration applications designed for application integration such as WebSphere Business Integration, Microsoft BizTalk, TIBCO, and BEA products. These provide an architecture and runtime environment that uses adapters to 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 can be viewed as a "super-bridging"-type solution where the bridge doesn't

connect system to system, but instead a series of bridges connect different systems to a single hub having its own messaging and communication format. These bridges also connect the hub back to the different systems so that translated messages can be passed to and fro between them, using the centralized hub as the transport mechanism. As such, it can be seen that from the point-of-view of management, it should fare about the same as the bridging methodology.

### 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. Architecturally, from a high level it appears to be very similar to EAI-style applications, and it is – with one major and important difference. With an Enterprise Service Bus the 'hub' on which all traffic flows is open, and standards-based, not proprietary. Therefore, connectors for different technology types can be chopped and changed between different ESB implementations. Also, when it comes to management, standard management platforms would be usable, as opposed to ones designed for the proprietary format of EAI applications.

The concept is interesting, and in its infancy, so ascertaining its manageability is difficult. However, one can deduce that as 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. Once new management platforms appear that can interface with the standard formats used by the ESB, its manageability characteristics should improve drastically. The closeness of ESB to Web Services from a data point-of-view should mean that they will evolve rapidly.

## 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 key management elements such as Fault Management, Configuration Management, Accounting Management, and Performance Management would therefore have 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. So interoperability between Mono and a WebSphere-based application can still be implemented and managed using technologies built on Mono such as Web Services.

### Platform Unification

The management of mixed-mode systems brings enough challenges of its own without adding more through an interoperation technology such as Web Services or bridging. Hence, the strategy of platform unification, where a single runtime stack is used, and many interoperability issues are eradicated. It should be noted that this is separate and distinct from *porting* where the application is rewritten at the source code level to move from one platform to the other.

There are two main initiatives around this area.

The first is J#.NET from Microsoft. The promise of J#.NET is that it lets Java developers take their existing Java source code, import it into a Visual Studio.NET IDE, and compile it into the MSIL language that will execute on the .NET runtime. The JDK dependencies have been implemented as .NET namespaces and, as such, dependencies and code won't have to be rewritten in many cases. For example, if the code implements Enterprise Java Beans, it will have to be re-implemented using a .NET-based technology such as System.EnterpriseServices. However, for many applications, the support offered by the J#.NET Framework may be enough to provide cost-effective platform unification.

The second is called Visual MainWin for J2EE, and it's published by Mainsoft. This provides platform unification in the other direction – allowing .NET server applications to be re-hosted on J2EE through the compile time conversion of C# or VB.NET code to Java bytecode. As such, .NET applications will interoperate with ones running on J2EE since they'll be running in the same JVM, and cross-process communications through bridging, Web Services and like aren't necessary. Thanks to the Open Source nature of Mono, Mainsoft used its technology to re-host the .NET framework on J2EE. Using Visual Studio.NET, through the Visual MainWin for J2EE plug-in, developers can code, compile, deploy, and debug their .NET applications on J2EE. Additionally .NET applications, as they are implemented as Java bytecode, can be integrated directly into Java applications, and Java assets such as EJBs can be consumed from ASP.NET applications without going through a bridging technology.

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 or .NET mechanisms, and are fully compatible with the environment and any existing management systems, such as Tivoli or Microsoft Operations Manager (MOM).
- **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. Application management and update deployment are also greatly simplified thanks to a uniform procedure on a single platform type. All resources can be managed uniformly through the appropriate management console.

- **Accounting Management:** As the underlying technology system is now standardized on a single one, integrating with accounting management for SLA and other metrics is straightforward.
- **Performance Management:** Performance in mixed-mode systems can actually increase as translation technologies such as bridging, EAI, or Web Services aren't necessary when using platform unification
- **Security Management:** Under the platform unification model, security is provided through a single platform, be it the .NET one (with J#) or the J2EE one (with Visual MainWin for J2EE). This homogenous model is used across the application and allows for managing users, authorization, and authentication in a single domain. This simplifies managing security, and eliminates the need to synchronize user and authorization stores across multiple domains. It also decreases the security risks of credential information being passed between domains and platforms on wire protocols such as HTTP or HTTPS. When hosting on J2EE 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.

So, although mixed-mode systems are common, attempting to manage data centers in which they are present is a challenging task. When evaluating the choices unify applications that need to interoperate with each other, think about how management is made more complex by the technology layers that are necessary to get systems to talk to each other. Platform unification, where .NET Framework-based applications are re-hosted on J2EE, or vice versa, reduces the need for interoperability layers and reduces overall complexity. ☛

*"Ithaca College needed thorough spam protection and the ability for users to choose their level of scanning through a simple, clear interface. CanIt-PRO offers both." – Lesli LaRocco, Systems Administrator, Ithaca College*



# Case Study



## About Ithaca College

- **Type:** Higher education
- **Number of mailboxes:** 7,000+

## E-mail Environment and Use

- Acts as "ISP" to students and staff
- E-mail: iPlanet messaging and directory services
- Server: UNIX box running Sun Solaris 8

## CanIt Results

- Thorough spam control for all e-mail users
- Minimal administrative involvement is required
- Individual users control their own levels of spam control
- Eliminated help-desk calls about spam and offensive e-mail content

## College Solves Spam and Virus Woes While Upholding Individuals' Rights and Freedoms

For a higher education provider, upholding individual rights and freedoms of speech, information and privacy can be a tangled – and sometimes contradictory – web. For Ithaca College of Ithaca, New York, the search for an anti-spam solution was guided first by the principle of finding one that end-users could control.

"We wanted end-users to have as much control as possible over their spam," confirms Lesli LaRocco, systems administrator at the college. "That's really important here, where there isn't a top-down decision-making policy about what is acceptable content." But, academic freedom aside, a barrage of pornographic spam caused such a furor among email users that a solution to block such offenses became necessary.

### Distributed Policy-Making

Ithaca College essentially required an anti-spam solution with policies that matched its own. LaRocco explains, "We have a general policy for the behavior of all users on the network but we only do things with the consensus of the community."

LaRocco's approach was to find a user-controlled product – a strategy that the college's various departmental liaisons agreed with.

"We looked at various anti-spam solutions, but all were very top-down, relying on a single source to decide what 'bad words' to filter out," LaRocco remembers. "Also, a lot of the other solutions had features that we just didn't need, such as tools to manage users' content and avoid corporate espionage."

After evaluating CanIt-PRO on a trial basis, Ithaca College implemented the product, which is now being rolled out the college's 7,000+ student and faculty. LaRocco says response has been excellent from both user and administrative perspectives.

### (So)Big Benefits

Just as valuable, says LaRocco, is the administrative time that CanIt-PRO has saved the college – such as when the SoBig virus attacked. "When the Blaster/SoBig worms hit our campus, CanIt-PRO was invaluable both in blocking and dumping them. During those weeks, our volume of mail was five times higher than normal, and being able to stop these messages at the gate was very helpful. Without CanIt-PRO, our anti-virus software would have stripped the virus and delivered the (useless) messages, meaning our users would have received hundreds of junk mails per day in some cases."

### Flexibility and Control: All in One

LaRocco says, "Implementing CanIt-PRO has been a snap. It was easily the smoothest product rollout we've ever had. And, we rarely get a complaint about false positives or missed spam."

"The simple GUI works really well for our users, as we have a real variety of levels of technological proficiency in this community. The GUI is so simple but it's also flexible – people who want granular control can have more complex choices, and the administrative interface is very well designed."

LaRocco has also found maintenance and customization of the system to be simple. The college customizes the CanIt-PRO web pages "to look and feel more like our other Ithaca College web pages. Customization has been very straight forward, and from an administrative perspective, this system is as easy to maintain as any other server product that we run – it's just a few minutes a day." ■



# IT'S THE END OF THE LINE FOR SPAM.



## PUT IT IN THE CAN.

Ads, fraud schemes, porn, viruses, phishing attempts...  
if it's offending e-mail, Roaring Penguin Software's CanIt™  
solutions will stop it before it reaches your mail server.



- Simple administration
- Automatic spam and virus updates
- Compatible with all e-mail servers
- Available as software or hardware appliance
- Flexible end-user controls

Throw the book at spam today!  
Request a free 20-day CanIt evaluation  
and stop spam on YOUR terms!

**ROARING**   
**PENGUIN**  
SOFTWARE INC.

# Blended Development

*Interview with Blake Connell, Director of Product Marketing for WebLogic Server, BEA Systems*

by Roger Strukhoff

**A**fter BEA's announcement regarding their release of the BlueDragon BEA WebLogic Server, SYS-CON Media had the chance to talk with Blake Connell.

**SYS-CON:** Tell us about your BlueDragon announcement.

**Blake Connell:** We're announcing and making available for download BlueDragon BEA WebLogic Edition, enabling the movement of existing ColdFusion and CFML applications onto a modern platform, BEA WebLogic Server. Our customers can now take existing applications as is (and) unmodified, a key benefit, and move them to a platform we know and love that is Java-compatible. They can get all the benefits such as failover, clustering, and advanced administration capabilities without having to rework their applications

**SYS-CON:** How will you be offering it? Is this being done through a license?

**Connell:** We're working with New Atlanta and are offering it on our Website for download, as well as supporting it directly. You can download it today to use for development purposes without any fees or restrictions. Then, if you put it into production, we have a per-CPU pricing model and support.

**SYS-CON:** How will your customers be able to reuse legacy code? How does this work?

**Connell:** It's fairly straightforward. You are literally repackaging your CFML applications as either EAR or WAR files. Your audience is familiar with enterprise archives and Web archives, and you simply wrap them up and they get to port on top of WebLogic Server, as any Java application would. It's then available for all of the advanced capabilities in WebLogic Server 9, such as failing over, ad-



Blake Connell  
BEA Systems

ministrative features – it will appear in the console – the management capabilities and the monitoring functionality. So again, you simply take an existing application and rewrap it in a standard format and deploy.

**SYS-CON:** What sort of legacy apps, in what sort of vertical markets are you addressing?

**Connell:** One of the interesting things about CFML and ColdFusion is that it predates Java and was really one of the first platforms to create dynamic applications on the Internet. So there's a massive installed base and large development community as well. The applications span just about every vertical and industry marketplace out there.

We have customers in government who have made this move, in large financial services firms who have used this combination of BlueDragon right on top of WebLogic Server, as well as large airlines and transportation companies. It really runs the gamut.

**SYS-CON:** Do you have any specific references you can talk about?

**Connell:** Yes, we've referenced an

application with the State of Georgia. This is a publicly available reference, and there are others.

**SYS-CON:** How does the Blue Dragon announcement fit into BEA's overall application development strategy?

**Connell:** One of the things we announced at JavaOne last year was this notion of blended development, which primarily means that we take the best of open source and commercial code, blend the two together, and provide support for the single entity. The BlueDragon announcement plays into this strategy in that, although it's not open source, we are taking a non-traditional Java architectural development environment and blending it with a modern platform, WebLogic Server.

It plays into the general notion of what we believe customers want, which is the ability to leverage frameworks that aren't necessarily developed through the Java Community Process with those that are.

**SYS-CON:** So you're not locked into a single way of doing anything. In this case, it seems you are blending two separate communities into a unified environment that lets customers port legacy apps.

**Connell:** That's exactly it and it allows investments that have been made in CFML and ColdFusion to be maintained. Being able to maintain those investments is important to our customers.

**SYS-CON:** And more information can be found where?

**Connell:** More information can be found at [www.bea.com](http://www.bea.com), then select the WebLogic product area, and you'll find the BlueDragon BEA WebLogic Edition product in that family, and be able to get information about how to download the software. ☛



**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](http://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](mailto:roger@sys-con.com)





# IT'S IN THERE SOMEWHERE

**MAKE ANSWERS TO PERFORMANCE PROBLEMS COME TO YOU.**



**OPNET Panorama** offers powerful analytics for rapid troubleshooting of complex J2EE/.NET applications. Panorama quickly identifies how application, web, and database servers are impacting end-to-end performance. With Panorama, you can pinpoint the source of a problem, so time and money aren't spent in the wrong places.

*The most successful organizations in the world rely on OPNET's advanced analytics for networks, servers, and applications.*

**[www.opnet.com/pinpoint](http://www.opnet.com/pinpoint)**

**OPNET®**  
Making Networks and Applications Perform™

**OPNET Technologies, Inc.** 7255 Woodmont Avenue, Bethesda, Maryland 20814 phone: (240) 497-3000 • e-mail: [info@opnet.com](mailto:info@opnet.com) • NASDAQ: OPNT

© 2006 OPNET Technologies, Inc. All rights reserved. OPNET is a registered trademark of OPNET Technologies, Inc.



by Ron Sigal  
and Tom Elrod

# Using Virtual Sockets to Fix Software Broken by Firewalls

*Without violating security*

A character on the TV show “L.A. Law” once said, “It was the 60s. Safe sex meant keeping the parking brake on.” In the early days of public networks security carried a similar sense of urgency. In fact, the network was its own best defense. One of the authors remembers accommodating a colleague by spending an afternoon plotting the hops, gateways, and contorted syntax necessary to send an e-mail from Brooklyn to Michigan. (How many computer scientists does it take to send an e-mail?)

Those unlamented days have given way to a world in which an early protocol like FTP which, in active mode, requires a data transfer connection from the server to an anonymous port on the client, seems either charmingly naive or shockingly promiscuous (depending, perhaps, on when you were born). Of course, to be fair, passive mode, in which the client makes the data connection, was introduced by 1973, but we’ll focus on the active mode as a case study in using virtual sockets to overcome, safely, the restrictions imposed by firewalls.



## JBoss Remoting

Virtual sockets and their relatives are provided by the Multiplex sub-project of the JBoss Remoting project. JBoss Remoting is a general-purpose framework for distributed method invocations and asynchronous callbacks, featuring pluggable transports,marshallers, and object serializers. It currently ships with eight built-in transports, fourmarshallers, and two serializers, and custom versions are easily integrated.

Two of the primary design goals of Remoting, which is envisioned as the unified communications framework for the JBoss Enterprise Middleware Suite (JEMS) of software products, including the JBoss Application Server, are (1) to support a variety of communication

models, and (2) to allow any combination of pluggable components to be exchanged for any other purely through configuration changes, leaving the application code untouched.

A variety of protocols are implemented at the transport level, each by a pair of descendants of the RemoteClientInvoker and ServerInvoker classes, which, as their names imply, run on the client and server, respectively. These pairs of classes work together to transmit a method invocation from the client to an application-specific handler on the server and return the handler’s response to the calling program on the client. Together they constitute an **invoker**. For example, the socket invoker is comprised of the SocketClientInvoker and SocketServerInvoker classes, and HTTPClientInvoker and HTTPServerInvoker make up the HTTP invoker. Callbacks are handled by the same components with an inversion of client and server roles.

## The Multiplex Sub-Project

A Remoting feature request, submitted by the JBoss JMS group, was for an invoker that supports multiplexed network connections, allowing the server to send callbacks to the client on a network connection initiated by the client. The request was motivated by the need to support asynchronous message consumption in the context of firewalls configured

to prohibit incoming connection requests on non-well known ports, thereby defeating a server’s ability to make callbacks. Although the request was just for an invoker with multiplexing abilities, a design decision was made to push multiplexing down to the socket level, which, since all of the other transports ultimately use sockets, extends the multiplexing feature to other transport protocols. From this feature request and design decision came the Multiplex sub-project.

## Public Multiplex Classes

From the user’s perspective, the core of the Multiplex subsystem is a set of classes derived from their counterparts in the java.net and javax.net packages, including a virtual socket, two server socket implementations, input and output streams, and factories for virtual sockets and virtual server sockets. The primary design goal was to make these classes as functionally similar as possible to their non-multiplexing parents, so that, for example, a virtual socket could be passed without disruption to a method expecting a real socket, and for most of the classes this goal is largely satisfied. For example, the VirtualSocket.getInputStream() method returns a multiplexing input stream that supports read timeouts. The main source of complexity arises in the treatment of server sockets, which we discuss in the next section.

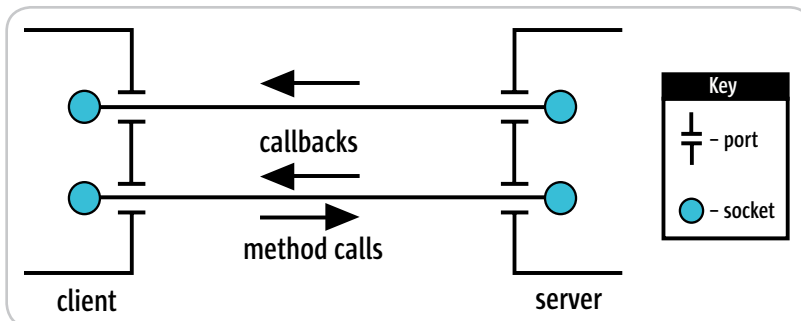


Figure 1 Method calls and callbacks with no port restrictions

Ron Sigal has taught mathematical logic and computer science at fine colleges and universities everywhere.

A co-author of “Computability, Complexity, and Languages, 2nd edition” (in use on five or more continents), he is proud to be a JBoss developer.

r.sigal@computer.org

## Virtual Socket Groups

For server sockets the situation is more complicated. Multiplexed connections are built using real sockets, so we need a server socket that can create real sockets. On the other hand, we need a server socket that can create virtual sockets. The classes `MasterServerSocket` and `VirtualServerSocket`, both derived from `java.net.ServerSocket`, fulfill these two roles, respectively.

Multiplexing a single TCP/IP connection between two actual sockets  $s1$  and  $s2$  is achieved by commingling the communications of multiple pairs  $(v1, v'1), \dots, (vn, v'n)$  of virtual sockets. The sets  $\{v1, \dots, vn\}$  and  $\{v'1, \dots, v'n\}$  are called **virtual socket groups**, where the members of one set are based on  $s1$  and the members of the other are based on  $s2$ . These sets are denoted  $G(s1)$  and  $G(s2)$ , respectively.

A virtual socket group is built in two stages, (1) creating the actual socket on which it's based and installing its first member, and (2) populating it with additional virtual sockets, and each stage is affected by the interaction of a virtual socket constructor with an instance of one of the two virtual server socket classes. The first stage involves the creation of a real socket and, accordingly, the `accept()` method of the `MasterServerSocket` inherits much of its functionality from `super.accept()`. When a virtual socket constructor finds no existing virtual socket group to join and it connects to a `MasterServerSocket`, the following sequence of events occurs:

1. The constructor creates an actual socket  $s1$  and connects it to the `MasterServerSocket`;
2. The `MasterServerSocket` accepts the connection, creates an actual socket  $s2$ , builds the infrastructure of a virtual socket group  $G(s2)$  based on  $s2$ , creates a virtual socket  $v'1$ , and adds  $v'1$  to  $G(s2)$ ;
3. The constructor builds the infrastructure of a virtual socket group  $G(s1)$  based on  $s1$ , creates a virtual socket  $v1$  connected to  $v'1$ , and adds  $v1$  to  $G(s1)$ .

Note that each connection to a `MasterServerSocket` results in the creation of a pair of new virtual socket groups. If the socket groups have to grow beyond their initial members, then one or both must be joined by a `VirtualServerSocket`, which is done by creating it and binding it to the port to which the underlying actual socket is bound. For example, suppose  $s1$  is bound to `bluemonkey.acme.com:6060` and  $s2$  is bound to `demo.jboss.com:8080`. Then executing `VirtualServerSocket vss = new VirtualServerSocket(6060); vss.accept();` on `bluemonkey.acme.com` will add a `Virtu-`

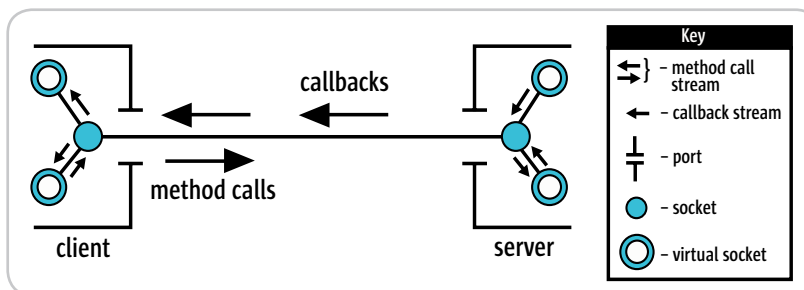


Figure 2 Method calls and callbacks in the Prime Scenario

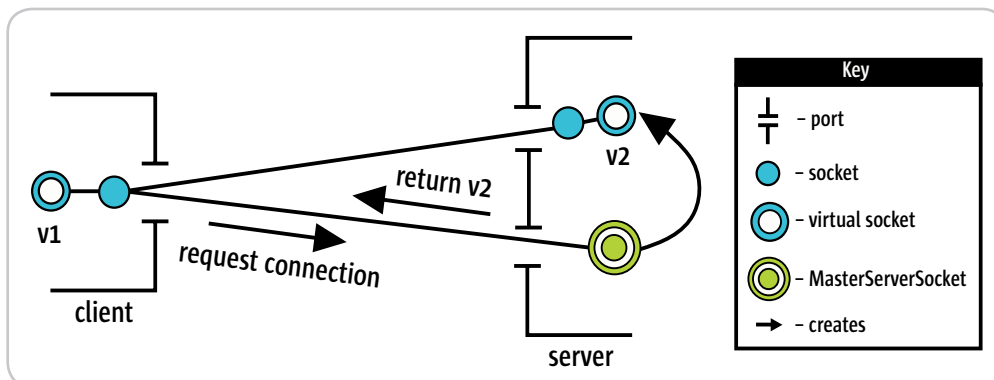


Figure 3 Setting up a synchronous connection.

alServerSocket to  $G(s1)$  and open it up for business. A subsequent constructor call `VirtualSocket v = new VirtualSocket("blue monkey.acme.com", 6060);` made on `demo.jboss.com` will add a new pair of communicating virtual sockets to  $G(s1)$  and  $G(s2)$ .

The difference between the two classes of server sockets may be expressed succinctly by categorizing their public methods.

### MasterServerSocket

The methods `accept()` and `toString()` are implemented directly. All other public methods are inherited except for `acceptServerSocketConnection()`, which is specific to `Multiplex`.

### VirtualServerSocket

The methods `accept()`, `bind()`, `close()`, `getSoTimeout()`, `setSoTimeout()`, `isBound()`, `isClosed()`, and `toString()` are implemented in the class. `getInetAddress()`, `getLocalPort()`, `getLocalSocketAddress()`, `getReceiveBufferSize()`, `getReuseAddress()`, `setReceiveBufferSize()`, and `setReuseAddress()` are delegated to the underlying socket. `getChannel()` returns null, and the methods `connect()` and `isConnected()` are specific to `Multiplex`. There is also a constructor with the signature `public VirtualServerSocket(InetSocketAddress remoteAddress, InetSocketAddress localAddress, int timeout)`, which allows a new `VirtualServerSocket` to bind and connect at the same time.

The striking thing here is the existence of the delegated methods and connection-oriented methods in the `VirtualServerSocket` class, both of which seem out of place for a server socket. In fact, they reveal the true, rather hybrid, nature of `VirtualServerSocket`, which uses a real socket to implement the behavior of a server socket. The important functional implication is that, like a socket, a `VirtualServerSocket` has a *connection to another host*, and so it can accept connect requests only from virtual sockets on that host.

The rules for creating and joining socket groups are spelled out in detail in the `Multiplex` documentation (see <http://labs.jboss.com/portal/jbossremoting/docs/multiplex/multiplex.html>), including a discussion of the **accidental connection problem**, which necessitates the constructors and methods that support binding and connecting in a single atomic action.

## The Prime Scenario

The typical application of multiplexing in a Remoting context is illustrated by the Prime Scenario in which a client requiring both synchronous and asynchronous responses from a server is behind a firewall blocking connections to anonymous ports. Without the firewall, we would have the situation in Figure 1, which requires no multiplexing. With the firewall we have to construct the Prime Scenario as in Figure 2.



Tom Elrod is a core developer for JBoss, Inc and lead of the JBoss Remoting project.

tom@jboss.com

Every Multiplex scenario begins with a MasterServerSocket that creates the underlying actual socket. Figure 3 illustrates the first step in creating the Prime Scenario, in which a virtual socket *v1* connects to a MasterServerSocket that creates and returns a reference to a new virtual socket *v2*.

In Figure 3 we have a connection between *v1* and *v2*, which can support synchronous communication but which offers nothing not provided by actual sockets. To create a second connection for callbacks, we add a VirtualServerSocket to the client's socket group. The second step in constructing the Prime Scenario is illustrated in Figure 4, in which the constructor (or factory method, which calls a constructor) is called on the server to create virtual socket *v3* to support callbacks. The constructor sends a connection request to the VirtualServerSocket on the client, which creates new virtual socket *v4* and sends a reference to *v4* back to *v3*. At this point we have a socket group on the client with two virtual sockets and a VirtualServerSocket, a socket group on the server with two virtual sockets, and the Prime Scenario is set up.

A sample client and a sample server for the Prime Scenario are given in Listing 1 and Listing 2 which can be downloaded from the online version of this article at <http://jdj.sys-con.com>.

### FTP Case Study

As a test of the usability of the Multiplex system, we modified an FTP client and an FTP server, both Open Source, so that an active mode data connection could be opened from the server to the client reusing the control connection made from the client to the server. The idea was to use existing code to see

how disruptive the necessary changes would turn out to be. For the client we chose the Apache Jakarta Commons Net library (<http://jakarta.apache.org/commons/net/>) and for the server we chose the Apache FTP server (<http://incubator.apache.org/ftpserver/index.html>). The Jakarta Commons Net library doesn't have a complete FTP client, just the classes necessary to build a client and a sample class, examples.ftp, which logs into an FTP server and transfers a single file, so that's what we used.

We made the changes in Listing 3 for the client. Note that red lines were deleted, green lines were inserted, and the yellow lines are particularly relevant to our discussion.

For the server, we made the changes shown in Listing 4 (Listings 4 and additional source code can be downloaded from the online version of this article at <http://jdj.sys-con.com>.)

And we can make the following observations about our experiment:

- Multiplex works. We were able to create a control connection and a data connection over a single shared TCP/IP connection.
- The changes were fairly minor and localized, restricted almost, though not quite, to swapping in virtual sockets and server sockets to replace their real counterparts. There were two variations on these changes.
  - Changing the client's server socket and the server's client socket involved adding classes and modifying code to replace explicit references to their respective classes.
  - Changing the server's server socket was more pleasant, since it involved adding a new class and changing a configuration file.

It should be noted that, had the code been more configurable, none of these changes would have required changes to existing code.

- The code change that configures the client to run in active mode would be unnecessary for a real FTP client, since the mode is settable at runtime by user command.
- The one code change that seems to be unavoidable in the current release is the one that requires the client's server socket to bind to the port used by the control connection, since that's how the VirtualServerSocket is told to join the existing virtual socket group.

### Conclusion

The Multiplex sub-project of JBoss Remoting provides multiplexing versions of various classes in the standard java.net and javax.net packages, allowing multiple virtual network connections to share a single real TCP/IP connection. Though they were developed to support a JBoss Remoting feature request, these classes can be applied independently and, in particular, they can be used to open connections to anonymous ports in the presence of a firewall while violating neither the letter nor the spirit of any security policy.

For this article we selected an Open Source FTP client and an Open Source FTP server and modified them slightly, replacing standard sockets and server sockets with their virtual counterparts. We found that if the target code is sufficiently configurable, the use of socket and server socket factories can almost eliminate the need to change existing code.

We found one necessary code change that can't be configured away. We can derive from this observation that a desirable feature for the next JBoss Remoting release would be default behavior that allows a VirtualSocket or VirtualServerSocket that binds to port 0 to join an arbitrary existing socket group. In the current case the new VirtualServerSocket would find exactly one existing socket group and join it, which is just what we want.

Which shows that there's no substitute for kicking the tires and taking it for a drive. ☺

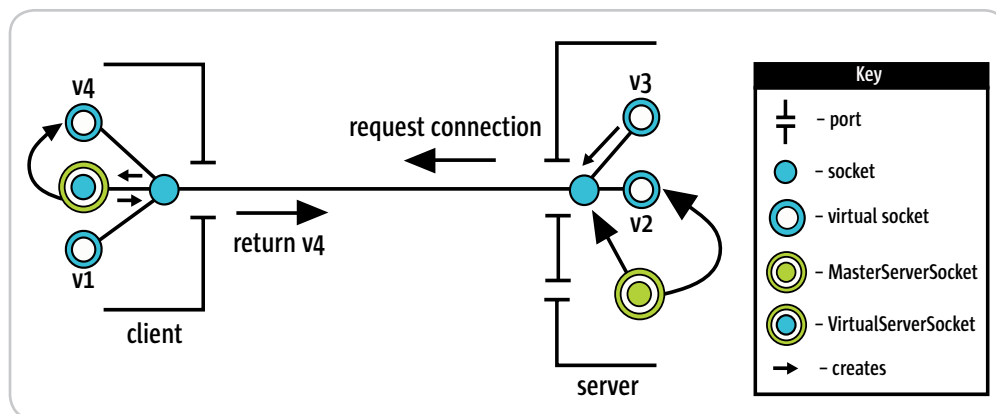


Figure 4 Adding an asynchronous connection to Figure 3



```

1. Add org.apache.commons.net.ftp.VirtualSocketFactory class:
package org.apache.commons.net.ftp;

import java.io.IOException;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.UnknownHostException;
import org.apache.commons.net.SocketFactory;
import org.jboss.remoting.transport.multiplex.VirtualServerSocket;
import org.jboss.remoting.transport.multiplex.VirtualSocket;

public class VirtualSocketFactory implements SocketFactory
{
    public Socket createSocket(String host, int port) throws
        UnknownHostException, IOException
    {
        return new VirtualSocket(host, port);
    }

    public Socket createSocket(InetAddress address, int port) throws
        IOException
    {
        return new VirtualSocket(address, port);
    }

    public Socket createSocket(String host, int port, InetAddress
        localAddr, int localPort)
        throws UnknownHostException, IOException
    {
        return new VirtualSocket(host, port, localAddr, localPort);
    }

    public Socket createSocket(InetAddress address, int port, InetAddress
        localAddr, int localPort)
        throws IOException
    {
        return new VirtualSocket(address, port, localAddr, localPort);
    }

    public ServerSocket createServerSocket(int port) throws IOException
    {
        return new VirtualServerSocket(port);
    }

    public ServerSocket createServerSocket(int port, int backlog)
        throws IOException
    {
        return new VirtualServerSocket(port, backlog);
    }

    public ServerSocket createServerSocket(int port, int backlog,
        InetAddress bindAddr) throws IOException
    {
        return new VirtualServerSocket(port, backlog, bindAddr);
    }
}

2. Derive org.apache.commons.net.ftp.FTPVirtualClient from org.apache.com-
mons.net.ftp.FTPClient and set socket factory to VirtualSocketFactory in
the constructor:

package org.apache.commons.net.ftp;

public class FTPVirtualClient extends FTPClient
{
    public FTPVirtualClient()
    {
        super();
        setSocketFactory(new VirtualSocketFactory());
    }
}

3. Modify examples.ftp to use FTPVirtualClient (and also to use active
mode):

package examples;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintWriter;
import org.apache.commons.net.ftp.FTP;
import org.apache.commons.net.ftp.FTPClient;
import org.apache.commons.net.ftp.FTPVirtualClient;
import org.apache.commons.net.ftp.FTPConnectionClosedException;
import org.apache.commons.net.ftp.FTPReply;

/**
 * This is an example program demonstrating how to use the FTPClient class.
 * This program connects to an FTP server and retrieves the specified

```

4. Change a call in `FTPClient` to bind the server socket to the same port used by the socket that transmits commands to the server:

# Maven: A Different Way of Looking at Software Development

by Jason van Zyl

## *A real-time engineering framework*

Software development is typically carried out in an opaque environment where progress can be slow and (too often) the resulting build processes lack visibility, transparency, and collaboration. The Apache Software Foundation's Maven project approaches the problem of building software by providing the technical underpinnings for a set of development methods that enable engineers and other stakeholders in a project to optimize build reliability, accelerate build velocity, and capture and share build knowledge. Maven differs from current script-based approaches to building software by first defining, standardizing, and then publishing the build process as a logically organized structured lifecycle. Having evolved from the Open Source development world of distributed, asynchronous, iterative, and highly component-based engineering, Maven's technology helps development teams effectively cooperate to create and deliver successful software projects consistently.

While people new to Maven tend to think of it as strictly a tool for builds (akin to Ant), Maven actually offers a set of tools, including a Project Object Model (POM), user-defined plug-ins, and functionally rooted best practices, whose goal is to measurably improve software development processes as a whole, rather than simply enable builds to compile.



But while there's definitely a "non-technical" element to developing software with Maven, its strong use of patterns and emphasis on community-oriented practices, the key to building with Maven comes from how its tooling technology actually enables and reinforces development best practices across a team. As anyone who has ever been part of a successful project knows, teams must creatively combine technical and non-technical elements to deliver quality projects. Maven's goal, in terms of its tooling, is to provide a solid and cohesive technical framework that can mitigate some of the human factors that hinder software development (e.g., one developer's code contribution breaking the build, non-standard

locations for relevant cross-related project resources, and the inability to integrate new tools into the build process). Moreover, once engineers are relieved of the more mundane, time-consuming aspects of the build process, they can get on with the important work of creating value at the application level.

So let's take a closer look at the Maven project and explore how, when combined with good internal build procedures, its use can improve overall software development processes:

- Apache Maven 2.0, the foundational POM-based build engine for Java projects
- Apache Continuum 1.0, the Continuous Integration server fully integrated with Maven
- Plug-ins, the standard way to work in Maven and extend its functionality
- Maven SCM, the abstraction layer for SCMs in Maven's plug-ins and in Continuum, which supports CVS and SVN, as well as Perforce, Subversion, and ClearCase
- Maven Wagon, the abstraction layer for general transport, heavily used in Maven and the Maven-Ant tasks library for moving artifacts between repositories
- Doxia, the content generation library used to generate project sites and create build reports
- Surefire, the abstraction layer for testing frameworks that

“Maven's real-time engineering framework supplements community-oriented best practices to measurably improve the predictably and velocity of software development and not just software builds”

As chief architect and co-founder of Mergere, Inc., **Jason van Zyl** focuses on improving the software development infrastructure associated with medium to large-scale projects, which has led to the founding of the Apache Maven project. He continues to work directly on Maven and serves as chair of the Apache Maven Project Management Committee.

support JUnit and general scripting (TestNG support is in development)

- Archetypes, a rapid project prototyping tool that quickly sets up a template Maven project

Typically, developers who come to Maven are already familiar with Ant, so there's a natural association made between the two projects. Ant is a script-based toolbox, and an excellent one at that. However, Maven isn't simply a build tool and, as such, it's not a replacement for Ant. Instead, Maven combines its POM, plug-ins, Continuum, and embedded build lifecycle phases (which can actually include Ant tasks) to offer engineers a software development infrastructure that delivers project visibility, re-usability, maintainability, and comprehensibility.

Originally Maven arose from a very practical need to make several Apache projects functionally build in the same way. Then, as now for most developers, being able to freely move between multiple

projects meant that each developer needed to clearly know and understand how each of the individual projects worked. This meant that for every project a developer needed to build, he had to repeatedly decipher and learn not only how to build the project in question, but also become familiar with its related testing, documentation, reporting, and deployment practices. Since all of the original Apache projects shared a number of build characteristics (lifecycle phases), Maven was developed to harness these commonalities. Given that all software projects need to be built, tested, packaged, documented, and deployed at the most base level – Maven was created to logically organize this build model and then orchestrate the process. Since there are also a great number of sub-functions possible at each phase of the build, test, package, document, and deploy lifecycle, Maven's build model lets users use (and create) plug-ins to meet build-specific requirements. But, at the same time, Maven constrains the overall build

process to a single recognizable and repeatable model, creating a build framework (versus a build tool) that saves developers from having to (re)learn unique project builds time and time again.

### Use of Patterns

Maven's build pattern emphasis was heavily influenced by the Gang of Four's design patterns that in turn borrowed heavily from Christopher Alexander's work in the realm of architecture. According to Alexander, "...patterns help create a shared language for communicating insight and experience about problems and their solutions." Maven's goal is to define a build structure so that problems can be approached consistently in terms of a common framework. By providing a lingua franca build, Maven facilitates build comprehension based on the following underlying framework concepts of:

- Convention over configuration
- Model-driven declarative execution
- Standard directory layout
- Build lifecycle

**DEVELOPERS,  
LOOK NO FURTHER**

### Highlights:

- Web Architecture Symposium  
Saturday, June 17
- Hands-on Training
- BI Summit
- More Than 100 Technical Presentations
- Built-in Time to Meet and  
Talk with the Experts

**ODTUG**  2006  
**KALEIDOSCOPE**  
**CLARIFYING THE DEVELOPER'S  
CHANGING UNIVERSE**

**June 17-21, 2006**  
**Wardman Park Marriott Hotel**  
**Washington, D.C.**

To obtain the special conference rate of \$175/night at the Wardman Park Marriott Hotel, call 1-800-Marriott by May 12, 2006. Attention government employees: Government rates are available, but conference room rates are lower.

### TOPICS:

Application Development • Traditional Tools • Business Intelligence • Methodology  
Development DBA • Professional Development • Non-Oracle Development Tools

**www.ODTUG.com**



Each of these concepts helps developers deal with their build infrastructure at a higher level of abstraction, allowing more effective and timely communication. Let's explore each of these concepts further.

### Convention over Configuration

One of Maven's central tenets is to provide logical default strategies for the most common build tasks, so that developers can avoid the mundane details. This notion is known as "convention over configuration" and has been popularized by the Ruby on Rails (ROR) community, specifically encouraged by ROR's creator David Heinemeier Hansson, who summarizes the notion as follows:

"Rails is opinionated software. It eschews placing the old ideals of software in a primary position. One of those ideals is flexibility. The notion that we should try to accommodate as many approaches as possible, that we shouldn't pass judgment on one form of development over another. Well, Rails does, and I believe that's why it works.

With Rails, you trade flexibility at the infrastructure level to gain flexibility at the application level. If you are happy to work along the golden path that I've embedded in Rails, you gain an immense reward in terms of productivity that allows you to do more, sooner, and better at the application level.

"One characteristic of opinionated software is the notion of "convention over configuration." If you follow basic conventions, such as classes are singular and tables are plural (a person class relates to a people table), you're rewarded by not having to configure that link. The class automatically knows which table to use for persistence. We have a ton of examples like that which all add up to make a huge difference in daily use."

Hansson articulates very well what Maven strives to deliver: using stan-

dard conventions saves time, makes it easier to communicate to others, and allows engineering teams to get to the important work of creating value in their applications faster and with less effort. With Maven, developers won't spend a lot of time getting a development infrastructure functioning, instead they "slot in" various pieces where required, while Maven takes care of the tasks (goals) for that lifecycle phase. Maven handles all the building, document generation, or deployment work through its POM.

### Model-Driven Declarative Execution

In Maven the POM is the central axis of work, since Maven is – by design – project-centric. Everything Maven needs to work with a given project is stored in the POM. Further, Maven uses the POM to execute and drive all of a project's build-related functions. Maven's model-driven execution relies on the POM (an XML-based metadata description of a project), which is stored in a standard location with its related project.

In short, the POM contains every important piece of information about your project and is the "one-stop shop" for finding anything related that project. Below is a simple example POM that contains the key elements common to all POMs:

```
<project>
<modelVersion>4.0.0</modelVersion>
<groupId>com.mycompany.app</groupId>
<artifactId>my-app</artifactId>
<packaging>jar</packaging>
<version>1.0-SNAPSHOT</version>
<name>Maven Quick Start Archetype</name>
<url>http://maven.apache.org</url>
<dependencies>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>3.8.1</version>
<scope>test</scope>
</dependency>
```

```
</dependencies>
</project>
```

The POM is typically stored in a file named pom.xml and sits at the top level of your project. The following are descriptions of the key elements in the POM above:

- **project** – the top-level element in all Maven pom.xml files
- **modelVersion** – indicates the version of the object model this POM is using (the model version itself changes infrequently but it's mandatory to ensure stability if/when the Maven community deems a model change necessary)
- **groupId** – indicates the unique identifier of the organization/group that created the project (groupId is one of the key identifiers of a project and is typically based on the fully qualified organization domain name, e.g., org.apache.maven.plugins is the groupId for Maven's plug-ins)
- **artifactId** – indicates the unique base name of the primary artifact being generated by this project, typically a JAR file (a typical artifact would have the form <artifactId>-<version>.<extension>, e.g., myapp-1.0.jar)
- **packaging** – indicates the package type to be used by this artifact (e.g., JAR, WAR, EAR, etc. and describes that the artifact produced is JAR, WAR, or EAR, but can also indicate the specific lifecycle to be used)
- **version** – indicates the version of the artifact generated by the project.
- **name** – indicates the display name used for the project (used in Maven-generated documentation)
- **url** – indicates where the project's site can be found (used in Maven-generated documentation)

“One of Maven's central tenets is to provide logical default strategies for the most common build tasks, so that developers can avoid the mundane details.”

- **description** – provides a basic description of the project (used in Maven-generated documentation)

### Standard Directory Layout

Another tenet Maven follows to simplify project development is the use of standard directory layouts for project sources, additional project resources, and documentation (generally referred to as “project content”). Maven encourages a common arrangement for project content so engineering teams can quickly and predictably find any project-related items whenever Maven builds projects. This tenet dovetails with Maven’s “convention over configuration” concept. So, while the location of the project content isn’t critical to a project’s success, it is critical that every member of that project team (both now and in the future) knows exactly where to look to find the project’s content. Maven’s standard directory structure ensures that developers can become familiar with a project tree and easily find project content simply by navigating that project’s Maven structure.

### Build Lifecycle

Recall earlier we said that while projects can have nearly infinite variability, Maven binds these potential inconsistencies and discrepancies in the well-defined structure of its build framework. The framework that Maven uses to accommodate project variations is defined as the build lifecycle. In Maven this framework consists of an ordered series of lifecycle “phases,” where each phase can perform one or more actions (a.k.a. “goals” in Maven-speak). The build lifecycle’s phases use highly abstracted names such as *compile*, *test*, *package*, and *deploy*. While any number of goals (actions) can be performed during the test phase, the framework for the lifecycle itself remains well-defined and clearly articulated for any developer working on the project. During testing, for example, an engineer may need to run unit tests, create databases, deploy EAR files to an application server, and run coverage tools. With Maven, there are no restrictions as to the number or type of goals required; there is simply a convention (an embedded best practice) that defines how, where, and when these actions will be processed — specifically in the test phase, as it’s defined in Maven’s build lifecycle.

### Visibility and Collaboration

Consider your own development environment, where everyone is constantly rooting around trying to figure out where all the different bits and pieces that make up a given project are located. This means there’s very little opportunity for the team, or any individual engineer, to comprehend the project as a whole. While Maven’s basic concepts of convention over configuration, model-driven declarative execution, standard directory layout, and build lifecycle all support build comprehensibility and maintainability, the next step in improving the software development lifecycle requires build visibility and reusability.

Without visibility, it’s unlikely that an individual engineer will ever know what another engineer has contributed. As such, there’s a good chance that useful code won’t be leverageable — forcing teams to lose out on the network effects of code reusability. Even a well-maintained build can end up being a silo if there’s no way to effectively share that project knowledge widely. As a result, Maven provides standard reports and a continuous integration server, Continuum, for the specific purpose of improving project visibility. Maven’s reporting capabilities, when used in conjunction with Continuum, provide real-time metrics regarding not only the overall health of any Maven-built project, but also fine-grained details about code coverage, code quality, developer activity, and project issues.

Maven’s goal of delivering project visibility, reusability, maintainability, and comprehensibility takes it beyond the confines of being a build tool. It ensures that organizations and teams can leverage not only all project intelligence, but also the ongoing health assessments of their projects. Maven’s real-time engineering framework supplements community-oriented best practices to measurably improve the predictably and velocity of software development and not just software builds. ☺

## Build interactive diagrams easier than you ever imagined

**C**reate custom interactive diagrams, network editors, workflows, flowcharts and design tools. For web servers or local applications. It's easy-to-use and very flexible. We're the first developer of diagramming components and still the best. Find out for yourself: download our FREE fully functional evaluation kit, with full support at [www.nwoods.com](http://www.nwoods.com).

**FREE Download With Full Support!**

**GoDiagram**

Interactive diagram components

[www.nwoods.com](http://www.nwoods.com)





**Joe Winchester**  
Desktop Java Editor



# Whatifitis: 'To Dwell in the Future and Forget About Today'

Some of the words I dread most in a meeting are: "What if?" They're fine in the present tense of "What if a user tries this option?" or "What if the database read fails mid flight?", but as soon as the future tense is introduced I begin to worry. "What if the database and middleware changes?" or "What if sometime soon we don't just have to run on PCs but need to work on mobile phones?" There is also the future future tense such as "What happens to the UI if the operating system is ported to run on a wrist watch?" or "What if one day the company merges with another whose corporate standard is MAC and SNA?"

My first IT job was building computer systems for Insurance Syndicates to allow them to manage their policies, collect premiums, and pay out claims to the less fortunate of their clients. It's a difficult business arena, with companies either losing or making a ton of money; our job was to provide them with systems to help stack the odds in favor of the latter. As well as supporting an existing 10-year-old application, we were all busy working on the next one, which was an entire rewrite. It suffered from all of the second-system syndrome features that usually plague such projects, where every feature, bell, and whistle that didn't get put into the first one gets incorporated.

What ultimately plagued and doomed the project was the "what if" over-engineering that went into its design. The business analysts tried to soothsay every possible scenario that would occur in the future, including some that would have required wholesale deregulation of the market and acts of parliament before they could ever become reality. Such schemes were happily programmed in with the foresight that if they were needed, a soft-coded switch could be thrown and the application would adapt itself in flight. Database rows had unused columns that, at the throw of a switch, would be activated onto users' screens and reports. The reality, of course, was that none of the scenarios that the analysts predicted ever occurred, and the amount of additional soft coding and layers of abstraction meant the software



was slow and bulky. We lost almost all our market share to the competition, which just built lean, focused software. Even when big changes did occur to the market, such as the introduction of the Euro as an accounting currency, this required huge changes to the application because it naturally wasn't one of the future scenarios that had been predicted.

It's not just analysts who suffer from "whatifitis" syndrome. Programmers are just as guilty of using this as the excuse to over-engineer designs and code. The first OO project I worked on had to talk to a relational database back end, and we set about tackling this by creating code that could read and write our business objects. After a while it became apparent that the same logic was being used over and over, so this was refactored into a common set of class libraries, and behold a persistence mapping framework was born. Management got very excited about this, believing we'd built some kind of IT Rosetta stone, and talked to our software vendor who brought in some consultants. They raised the possibility that the database tables beneath our objects might change shape and we had to code for that possibility, as well as the possibility that our company might merge with another with a totally different database vendor and schema, and we should cater for that. There was a lot of buzz at the time about OO databases that just stored objects right onto disk without messy rows and tables, so to be ahead of the game our persistence layer catered to this. More and more layers of indirection were built into the code that now sat between the top-level objects and the da-

tabase reads and writes. This, by now, had become soft coded to look up in dictionaries and maps to figure out what it was actually supposed to be doing at each decision point. This was a cool idea that would allow a system administrator to merely tweak the soft-coded rules and introduce wholesale change to how and where the beautiful and pure OO system would get and put its data. The system never actually shipped, because after having spent about three years building it, when the company did actually merge with another they decided it was silly to spend so much on our project; they would rather spend time patching the existing system to deal with the challenges resulting from the merger.

I'm fortunate enough now to work for a major IT vendor, yet during a recent presentation was given a reminder that "whatifitis" is still alive and well. This occurred while showing a customer a product that lets them write their code once and have it run on several different clients: green screen, Java Swing, and HTML. The business scenario is for someone who has users of all three interfaces, and while the Web or Java interface we generate isn't going to win usability awards against bespoke AJAX or Swing code, it's a good solution for companies who have hundred of screens and just don't want the hassle of many expensive bespoke interfaces. Halfway through the demo, one of the customer's architects asked, "What about the X-Box?" He'd read an article that said how games consoles were the user interface of the future and wanted to know on which release were we going to support this feature. Meeting customers is always odd because I usually only get to be in a room full of managers and their high-level architects, rather than their coders who actually get the work done. Instead of replying, "What if you stopped worrying about the future and, instead of reading airport newsstand IT magazines full of FUD, you spent some time with your users, solving their day-to-day business problems"; I gave a wishy-washy answer that didn't break my company's business conduct guidelines. Pity, what if...? ☹

**Joe Winchester** is a software developer working on WebSphere development tools for IBM in Hursley, UK.

joewinchester@sys-con.com





*Ensure Interoperability.*

*Validate functionality.*

*Eliminate security vulnerabilities.*

*Test performance & scalability.*

*Confirm compliance.*

*Collaborate and reuse.*

# Ensure Secure, Reliable Compliant Web Services

 **PARASOFT**  
**SOAtest™**

As enterprises adopt Service Oriented Architectures (SOA) to deliver business critical data, ensuring the functionality and performance of Web services becomes crucial. Complex web services implementations require the means to thoroughly validate and test them to assure they are truly production ready.

Parasoft SOAtest is a comprehensive, automated tool suite for testing web services and complex Service Oriented Architecture (SOA) solutions to ensure they meet the reliability, security and performance demands of your business. SOAtest provides a total and holistic testing strategy for your SOA implementations including automated unit testing, graphical scenario testing, scriptless performance/load testing, security penetration testing, standards validation, message authentication, and more.

**If you are building serious web services, you need SOAtest. For more information regarding Parasoft SOAtest, call 888-305-0041 (x-3501).**

**Download a copy of SOAtest for a free evaluation today at [www.parasoft.com/WSJmagazine](http://www.parasoft.com/WSJmagazine)**

**Parasoft SOAtest clients include: Yahoo!, Sabre Holdings, Lexis Nexis, IBM, Cisco & more.**

 **PARASOFT**  
**We make software work.™**

**Automated Error Prevention™**

Parasoft Corporation, 101 E. Huntington Dr., Monrovia, CA 91016. For information, call 888-305-0041 (x-3501). Copyright ©2006 Parasoft Corporation. All rights reserved.  
All Parasoft product names are trademarks or registered trademarks of Parasoft Corporation in the United States and other countries. All other marks are the property of their respective owners.

BY NOW THERE ISN'T A  
SOFTWARE DEVELOPER ON EARTH  
WHO ISN'T AWARE OF THE  
**COLLECTION OF PROGRAMMING  
TECHNOLOGIES KNOWN AS AJAX!**

REAL-WORLD  
**AJAX**  
SEMINAR



**April 24, 2006**  
San Jose, CA



**June 5-6, 2006**  
New York, NY



**Oct 3-4, 2006**  
Santa Clara, CA

REAL-WORLD  
**AJAX**  
SEMINAR

How, in concrete terms, can you take advantage in your own projects of this newly popular way of delivering online content to users without reloading an entire page?

How soon can you be monetizing AJAX?

*This "Real-World AJAX" one-day seminar aims to answer these exact questions...*

**HURRY!**  
**LIMITED SEATING**  
**THESE SEMINARS WILL**  
**SELL-OUT**  
**CALL 201-802-3022**  
**TO REGISTER!**

Led by "The Father of AJAX" himself, the charismatic Jesse James Garrett of Adaptive Path, "Real-World AJAX" has one overriding organizing principle: its aim is to make sure that delegates fortunate enough to secure themselves a place – before all seats are sold out – will leave the seminar with a sufficient grasp of Asynchronous JavaScript and XML to enable them to build their first AJAX application on their own when they get back to their offices.

**Jeremy Geelan**  
Conference Chair, Real-World AJAX  
jeremy@sys-con.com

Register at...  
**www.AJAXSeminar.com**

**April 24, 2006** (San Jose, CA) ..... **\$1,195**  
See website or call for group discounts

**June 5-6, 2006\*** (New York, NY) ..... **\$1,295**  
See website or call for group discounts

**October 3-4** (Santa Clara, CA) ..... **\$1,495**  
See website or call for group discounts

**On-Deman Online Access** (Any Event) ..... **\$695**  
Purchase both New York and San Jose seminars for on-demand access

\*Includes SOA Web Services Edge plus  
Enterprise Open Source Conference & Expo



**LIVE SIMULCAST!**  
AROUND THE WORLD ON SYS-CON.TV

SPONSORED BY



# Featuring... Real-World AJAX Rock Stars!



**Jesse James Garrett (San Jose April 24)**  
*Father of "AJAX" Who Coined the Term in 2005*

Jesse James Garrett is the Director of User Experience Strategy and a founding partner of Adaptive Path, the world's premier user experience consulting company. He is author of *The Elements of User Experience* (New Riders), and is recognized as a pioneer in the field of information architecture. Jesse's clients include AT&T, Intel, Crayola, Hewlett-Packard, Motorola, and National Public Radio. Since starting in the Internet industry in 1995, Jesse has had a hands-on role in almost every aspect of Web development, from interface design and programming to content development and high-level strategy. Today, information architects around the world depend on the tools and concepts he has developed, including the widely acclaimed "Elements of User Experience" model. He is co-founder of the Information Architecture Institute, the only professional organization dedicated to information architecture. He is also a frequent speaker and writer whose work has appeared in numerous publications, including *New Architect*, *Digital Web*, and *Boxes and Arrows*.



**Adam Bosworth (San Jose April 24)**  
*Vice President of Engineering, Google*  
*One of the Fathers of XML & the Creator of MS Access*

Adam Bosworth is Vice President of Engineering, Google. He joined Google in 2005 from BEA Systems, where he was Chief Architect & Senior VP of Advanced Development. Prior to joining BEA, Bosworth co-founded Crossgain, a software development firm acquired by BEA. Known as one of the pioneers of XML, he previously held various senior management positions at Microsoft, including General Manager of the WebData group, a team focused on defining and driving XML strategy. While at Microsoft he was also responsible for designing and delivering the Microsoft Access PC Database product and assembling and driving the team that developed the HTML engine of Internet Explorer 4.0.



**Dion Hinchliffe (San Jose April 24)**  
*Cofounder & CTO, Sphere of Influence Inc.*  
*Editor-in-Chief, Web 2.0 Journal*

Dion Hinchliffe, newly appointed editor-in-chief of SYS-CON's pioneering Web 2.0 Journal, is cofounder and chief technology officer for the enterprise architecture firm Sphere of Influence Inc., in McLean, Virginia. A veteran of software development, Dion works with leading-edge technologies to accelerate project schedules and raise the bar for software quality. He is highly experienced with enterprise technologies and he designs, consults, and writes prolifically. Dion actively consults with enterprise IT clients in the federal government and Fortune 1000. He is a frequent speaker on AJAX, Web 2.0 and SOA and is currently the top-read SYS-CON.com blogger.



**Christophe Coenraets (San Jose April 24)**  
*Senior Technical Evangelist, Adobe*  
*AJAX/Flex Integration Guru*

Christophe Coenraets currently works as a Senior Technical Evangelist at Adobe. Before joining Adobe, Christophe was an evangelist at Macromedia, focusing on Rich Internet Applications and Enterprise integration. Prior to Macromedia, Christophe was the head of Java and J2EE Technical Evangelism at Sybase, where he started working on Java Enterprise projects in 1996. Before joining Sybase in the US, Christophe held different positions at Powersoft in Belgium, including Principal Consultant for PowerBuilder, and Manager of the Professional Services organization. Before joining Powersoft, Christophe worked as a developer and architect on several retail and BPM projects. Christophe has been a regular speaker at conferences worldwide for the last 10 years.



**Paul Rademacher (San Jose April 24)**  
*Google, Creator of HousingMaps.com*

Paul Rademacher is the creator of HousingMaps.com, which combined Craigslist and Google Maps for the first web mashup. Paul holds a Ph.D. in Computer Science from UNC-Chapel Hill, and worked as an R&D Engineer at Dreamworks Animation on such movies as *Shrek 2* and *Madagascar*. Since creating HousingMaps, Paul is now at Google.



**Jouk Pleiter (San Jose April 24)**  
*Co-Founder & CEO of Backbase*

Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur; he founded three successful software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market. Jouk previously was part of the founding team at the Interactive Agency Twinspark where he grew the company to a leading market position in Europe and was instrumental in the sale of Twinspark to Agency.com. He has an MBA from the University of Groningen.



**Kevin Hakman (San Jose April 24)**  
*Director of Product Marketing for TIBCO*  
*General Interface TIBCO Software*

Kevin Hakman is the director of product marketing for TIBCO General Interface, the award winning AJAX and Rich Internet Application framework and toolkit. Kevin Hakman pioneered AJAX in the enterprise co-founding General Interface in 2001. Since that time General Interface (aka "GI") has been powering Web applications that look, feel and perform like desktop applications, but run in the browser at Fortune 500 and U.S. Government organizations. General Interface was also the first to use its own toolkit to provide full visual tooling for AJAX when it released its 2.0 Version in 2003. TIBCO acquired General Interface in 2004 to extend its vision for service oriented applications to the end user. Kevin is a contributor to the SOA Web Services Journal and the AJAX Developer's Journal.



**Shanku Niyogi (San Jose April 24)**  
*Product Unit Manager of the UI Framework and Services Team Microsoft Corporation*

Shanku is Product Unit Manager of the UI Framework and Services (UIFX) team, which is responsible for delivering high-productivity UI framework technologies for the .NET platform, including ASP.NET, Atlas, Windows Forms, and frameworks for smart clients. Prior to his current role, Shanku was Group Program Manager of the Web Platform and Tools team on the Whidbey release of ASP.NET and Visual Web Developer. Shanku joined Microsoft in 1998 as a developer, having spent several years shipping products in the Windows ISV industry. Shanku holds a Bachelor of Mathematics degree in Computer Science from the University of Waterloo.



**Coach Wei (New York June 5-6)**  
*Chairman, Founder and CTO, Nexaweb*  
*The Creator of First Commercial AJAX Applications*

Coach Wei combines in-depth IT industry expertise with extensive education and research experience at MIT to drive technology innovation and business direction for Nexaweb. He founded Nexaweb in 2000 and served as CEO until summer 2003. Before founding Nexaweb, Coach architected and designed enterprise software for managing storage networks at EMC Corporation. As a graduate researcher at MIT, Coach developed software and hardware systems for non-destructive evaluation as well as signal/image processing algorithms. Coach was a finalist in the 1999 MIT \$50K entrepreneurship competition and holds several U.S. patents. An accomplished writer and speaker, Coach has published numerous articles on topics including: AJAX, J2EE and .NET, RIA development, XML, signal/image processing, composite materials and ultrasonic imaging. He has spoken at top industry events, such as JavaOne and Web Services Edge. Coach holds an MS in information technology from MIT.



**Ajit Jaokar (New York June 5-6)**  
*CEO, futuretext*  
*Author, "Mobile Web 2.0"*

Ajit Jaokar, based in London (England), is the CEO of a publishing company, futuretext (www.futuretext.com). He is currently writing a book about Mobile Web 2.0 (Mobile Web 2.0: The Innovator's Guide to Developing and Marketing Next Generation Wireless / Mobile Applications). Ajit also chairs Oxford University's Next-Generation Mobile Applications Panel and, since January 2006, has been a member of the Web 2.0 Workgroup. In his "Real-World AJAX" conference session, Ajit will discuss the "AJAX Use in Mobile Applications" as part of the wider impact of Web 2.0, sometimes referred to as the "Global SOA."



**Jonas Jakobi (New York June 5-6)**  
*AJAX Evangelist and Co-Author, "Ajax and JSF: Friend or Foe?"*  
*Jonas will autograph a copy of his book for all delegates!*

Jonas Jakobi is a principal product manager and evangelist for Oracle's Java/J2EE tool offering, JDeveloper, and over the past three years has been responsible for JavaServer Faces, Oracle ADF Faces, and Oracle ADF Faces Rich Client development features within Oracle JDeveloper. Jonas has been in the software business for 15 years. Prior to joining Oracle, he worked at several software companies in Europe, covering many roles including support, consulting, development, and project team leadership. Jonas' new book "Ajax and JSF: Friend or Foe?" released by Apress on February 25, 2006.



**John Fallows (New York June 5-6)**  
*AJAX Evangelist and Co-Author, "Ajax and JSF: Friend or Foe?"*  
*Jonas will autograph a copy of his book for all delegates!*

John Fallows, former lead developer for Oracle ADF Faces Rich Client, has been working in distributed systems for over a decade. After five years spent focused on designing, developing the JavaServer Faces standard to provide AJAX functionality, playing a leading role in the Oracle ADF Faces team, he recently joined an AJAX start-up. Originally from Northern Ireland, John graduated from Cambridge University in the United Kingdom and has worked in the software industry for more than ten years. Prior to joining Oracle, he worked as a research scientist for British Telecommunications Plc.



**Steve Benfield (New York June 5-6)**  
*Well-known AJAX Evangelist and CTO of Agentis Software*  
*Steve's first talk on "Aspect-Oriented Programming & AJAX"*

Steve Benfield is CTO of Agentis Software and one of the pioneers of AJAX technology, a gifted writer and a technical visionary. A technology marketer and strategist with 20 years of software entrepreneurship experience, a combination of qualities that made him the perfect choice of editor-in-chief for SYS-CON Media's inaugural publication 12 years ago. Steve's proven ability to determine marketing and technology strategies that align with market needs led to successful stints at SilverStream, where he started as technology evangelist and ended as CTO, and at ClearNova, an open source AJAX company, where he was CTO and AJAX evangelist.



**Jeremy Geelan**  
*Conference Chair*  
*Group Publisher & Editorial Director, SYS-CON*

Jeremy Geelan is group publisher and editorial director of SYS-CON Media, and is responsible for all print titles and online i-technology portals for the firm. He regularly hosts SYS-CON.TV, is executive producer of the "Power Panels with Jeremy Geelan" iTV series, and represents SYS-CON at conferences and trade shows, speaking to technology audiences both in North America and overseas. His i-Technology Blog is at [jeremy.linuxworld.com](http://jeremy.linuxworld.com) and he is conference chair of the upcoming iTvCon - "Internet TV Conference & Expo 2006".



## What you'll walk away with...

A New understanding of AJAX and how to make it work for you, plus:

- A) **Conference Proceedings Binder**  
Full-conference proceedings in a self-contained commemorative binder.
- B) **AJAX in Action Book**  
A copy of the Best-Selling AJAX book by Dave Crane.
- C) **Real-World AJAX "Secrets of the Masters" Book + DVD**  
New AJAX book edited by Dion Hinchliffe (Release Date: Summer 2006)
- D) **Notebook**  
Executive style keeps you looking sharp for note-taking wherever your business might take you!
- E) **Computer Back-Pack**  
A hip new way to conceal your laptop! The Urban Wonder Compu-Pack is an ingenious creation - a 600-denier polycanvas and nylon knapsack!
- F) **T-Shirt**  
100% preshrunk 6.1 oz. heavyweight cotton tagless shirt features shoulder-to-shoulder tape, and double-needle sleeves and bottom.
- G) **Porcelain Mug**  
Lightweight and durable, you can use it as a teacup, coffee mug or keep it as a pencil holder.
- H) **Satin Silver Contemporary Pen**  
Satin-silver plastic barrel with the look and feel of brass. Rubber comfort grip improves writing control and comfort.
- I) **Real-World AJAX Seminar on DVD**  
Watch complete seminar video and slide presentations (Release Date: Summer 2006)

For more information...

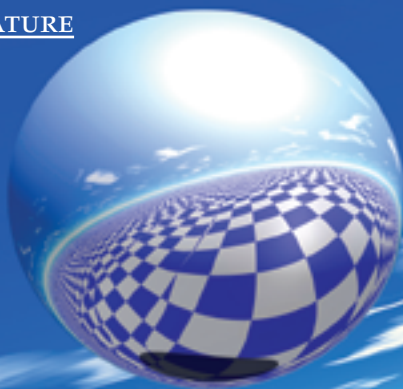
Call 201-802-3022 or email [events@sys-con.com](mailto:events@sys-con.com)



For more great events visit [www.EVENTS.SYS-CON.com](http://www.EVENTS.SYS-CON.com)

NOTE: SPEAKER LINE-UP SUBJECT TO CHANGE WITHOUT NOTICE  
VISIT [WWW.AJAXSEMINAR.COM](http://WWW.AJAXSEMINAR.COM) FOR THE MOST COMPLETE UP-TO-DATE INFORMATION





# Graph3D

Visualizing data  
using Java3D

by Valor Dodd

In today's work environment analyzing large amounts of varying data types is paramount. Graphing techniques can be an invaluable tool to understanding and interpreting that data. In many cases two-dimensional graphs, such as XY, scatter, pie, and bar charts, are sufficient. But increasingly more complex graphing techniques are needed. In these instances Java3D is an excellent resource with numerous features that allow personalized generation of three-dimensional data displays. Not only will Java3D yield better insight into the data by highlighting important aspects of the data, but it also makes attractive displays to spice up any presentation.

## Java3D

Before going into a detailed description of the Graph3D application and its execution, a basic understanding of Java3D is necessary. Java3D provides a rich set of functionality for viewing, constructing and manipulating geometric objects. Describing all the capabilities of Java3D is beyond the scope of this article, but it includes functionality for light modeling, solid and surface generation, object transformations, texturing and 3D text.

In Java3D, objects and transformations are built using a hierarchical tree structure as illustrated in Figure 1. The following are key classes of the tree.

- **Simple Universe** – The top of the tree is the Simple Universe. It creates the default viewing parameters for the 3D display. In most instances, the user doesn't perform operations with the Simple Universe except to make it the root of the hierarchical tree.
- **BranchGroup** – A BranchGroup is a tree node that holds logical groupings of objects. They are used to hold other nodes or to start a new tree branch.
- **TransformGroup** – A TransformGroup is a tree node that can apply transformation matrices to its children. TransformGroups can contain multiple BranchGroups and other TransformGroups for complex transformations. A class that's added to TransformGroups is the Transform3D class. It's used to hold transformation matrices and is added by using the setTransform() method. Transform3D will transform all the children of the TransformGroup. It also has helper methods to create matrices for rotations, scaling, and translations.
- **Leaf** – A Leaf node is where most of the objects are defined, such as lines, text, surfaces, and solids. The Leaf class is an

abstract class and as such cannot be instantiated. It also has no children. Instead, the Shape3D class (a subclass of Leaf) is used. Shape3D also include attributes for coloring, polygon fill, transparency, and material type. If the object to be drawn is a sphere, cylinder, box, or cone, Java3D has special classes that do much of the work for you. These special classes are subclasses of Shape3D. You can easily add textures to a Shape3D object. Textures are images that are draped over surfaces or solid models. For example, a sphere can be wrapped with an image of the earth to form a globe. Shape3D instances are added to BranchGroups or a TransformGroup for transformation.

Besides the tree classes, all Java3D programs must use a Canvas3D. It's a member of the java.awt(AWT) package and a Graphical-User-Interface(GUI) component that's usually added to a Frame or a javax.swing.JFrame. It's where the 3D rendering is done and has methods that set the eye viewing positions. The Canvas3D coordinate space defaults to a range of -1.0 to 1.0 for each axis. The center of the scene is 0,0,0 with the x-axis horizontal, y-axis vertical, and the positive z-axis coming out of the computer terminal.

Java3D includes three mouse behavior classes (MouseRotate, MouseZoom, MouseTranslate) that are used to rotate, zoom, and translate the display. They automatically compute and apply the transformation matrices to the TransformGroup specified in the constructors for the mouse behaviors. On a two-button mouse, MouseRotate is activated by depressing the left button and dragging the cursor in a horizontal (y-axis rotation) or vertical (x-axis rotation) motion. MouseTranslate is activated by depressing the right mouse button and dragging the cursor left and right. Activating MouseZoom is done by holding down the ALT button while dragging the left mouse up and down. This zooms the display in or out.

Light sources are used to realistically model solid objects. Java3D light source options include PointLight, SpotLight, Directional, AmbientLight, or any combination. They can be added as children to a BranchGroup. If this is done, the whole BranchGroup will be rendered with these light sources. For the Graph3D application, a separate BranchGroup was made to hold the light sources. The BranchGroup was then added as a child to the BranchGroup known as "objRoot." This was done so that the light sources could be easily detached from the Java3D hierarchical tree to make a wire frame display of the 3D graph.



**Valor Dodd** is currently a senior software engineer at Lockheed Martin in Denver. He is a certified Java Programmer and has more than 25 years of software experience developing computer graphics and GUI applications for the telecom and aerospace industries.

The definition of the light sources and how they are used is a complex subject and additional references should be consulted.

For a more complete description of the tree and on getting started with Java3D, an excellent Java3D tutorial can be found at [java.sun.com/developer/onlineTraining/java3D](http://java.sun.com/developer/onlineTraining/java3D).

## Graph3D Description

The Graph3D application is a program that plots three-dimensional data from a text file as illustrated in Figure 2. It is comprised of eight classes: Graph3D, FileIn, GraphData, BoundaryBox, Axis, Labels, Graph, and Grid. Included with the source code for Graph3D are JApplet1.java and Standalone.java. JApplet1 enables Graph3D to run from a web browser and Standalone enables it to run as a standalone application.

The following is a description of the sequence of events when the Graph3D application is executed. Both JApplet1 and Standalone pass an instance of a GraphData object into the constructor of Graph3D.java. GraphData reads in the data from a file using FileIn.java. Inside of Graph3D.java the SimpleUniverse is created and the top-level BranchGroup, called objRoot, is created with a call to the method createSceneGraph(). This method creates a BranchGroup with a TransformGroup as a child. Inside of the createSceneGraph() method a TransformGroup, called objTrans, is created and contains a Transform3D object that will handle the transformation matrix that applies to the whole scene. Three mouse behaviors (MouseRotate, MouseZoom, MouseTranslate) are added to objTrans. These behaviors allow for the automatic computation and application of the transformation matrix to objTrans. For the mouse behaviors to take effect, objTrans must set the appropriate capabilities. Use ALLOW\_TRANSFORM\_READ and ALLOW\_TRANSFORM\_WRITE in the setCapability() method of TransformGroup to allow the transformation matrix to be read and written. Unfortunately, the default behavior of Java3D doesn't let the matrix be read and written. In Java3D, many capabilities that a user would expect to have as a default must be explicitly set using setCapability() methods.

Next, the graph labels, axes, grid, and surface are generated via the classes Labels, Axis, Grid, and Graph. These are all subclasses of the Shaped3D class and are added to objTrans. All children of objTrans will be transformed using the mouse to transform the display. And finally a BranchGroup is added to objRoot called lightRoot that will handle the lights to be applied to the scene. The lights are made via a call to the makeLight() method in Graph3D.java. AmbientLight and two PointLight sources are used to shade the scene.

The following describe the vital functions:

- **Graph.java** does the 3D surface construction of the data to be plotted. The data is stored in the GraphData object as a two-dimensional array and are converted into a Point3f array so that they can be used by the QuadArray class. Any solid modeling that requires shading must be defined in a QuadArray or a TriangleArray. These two classes let surfaces be created using a grid of rectangles and triangles. Both the QuadArray and the TriangleArray also need an array of surface normals passed into their constructors. All shading algorithms require the surface normals of each individual quad or triangle so that they can apply the appropriate color. Other attributes of the data, such as transparency, material, and polygon fill are also set in the Graph class.
- **Grid.java** takes the QuadArray formed in the Graph class as an input to its constructor. Grid takes the QuadArray and draws it as lines instead of filled polygons as in the Graph class. The grid attributes are also set in this class. In Grid.java, you'll notice a call method setPolygonOffset(). Polygon offset is a necessary capability that must be used to display the line grid on top of the filled surface. Since the grid data points have the

same vertices as the filled polygon surface, both can't be displayed at the same location or seen unless the surface is translucent. Basically it "nudges" the line grid to a slightly different coordinate so the vertices of the grid can be seen.

- **Axis.java** creates the 3D axis lines. All the vertices of each axis are put into a LineArray that's used for the geometry of this Shape3D class. Note that our example data has only positive y-axis values. If your data have negative values this class will have to be augmented.
- **Labels.java** constructs the labels needed for the three axes. This class isn't a subclass of the Shape3D class. It makes a separate Shape3D object for each of the labels and for the graph title. Notice that the labels are added as children to a separate TransformGroup. This is necessary so that the labels can be scaled properly. The z-axis labels also need to be children of another TransformGroup so that they can be rotated along the direction of the z-axis. Each label is made with the Text3D class.
- **BoundaryBox.java** encloses the 3D graph with the three-sided translucent box. It's a BranchGroup that contains three BranchGroups each with a TransformGroup as a child. Each TransformGroup contains a Shape3D object that is a side of the translucent box. This configuration is necessary because each side of the box must be translated to the correct box side location.

As mentioned above, Java3D makes use of a Canvas3D object to do its final display rendering. For inexperienced Java GUI programmers, Canvas3D has features that can be problematic. Graph3D uses javax.swing (Swing) components because of their ability to programmatically change their appearance, but the GUI display can behave strange when mixing classes from AWT and Swing. For example, the AWT GUI component (Canvas3D) will always be drawn on top of Swing GUI components. This undesired effect is most noticeable when dragging a JFrame or a JInternalFrame

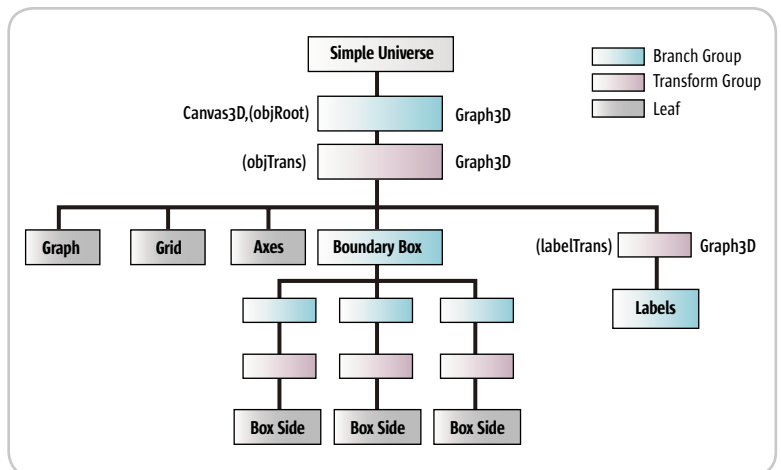


Figure 1 Java3D hierarchical tree

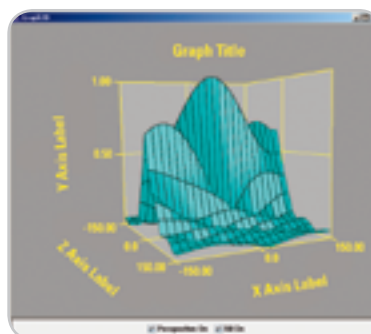


Figure 2 Example output from Graph3D

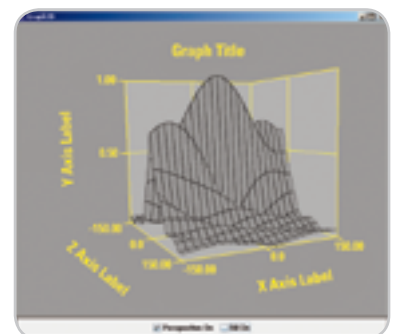


Figure 3 Graph3D wire frame

(Swing windows) in front of the window containing the Canvas3D. To mitigate this problem you can either rewrite Graph3D using only AWT components (a fairly simple task) or arrange your GUI display so that Swing components will never interfere with the Canvas3D. This feature has been reported to Sun Microsystems but fixes have yet to be released.

### Input Data

The file plot3D.dat (included with the source code) shows an example of the data format. The top part of the file describes the graph labels and data bounds. The data is formatted so that each coordinate is a corner of a square. The squares form a grid that has the same number of rows as columns. Of course, the format of the data file can be changed by simply changing the readData() method in FileIn.java to reflect the change. The data are read into a two-dimensional array that will be used by the Graph class in constructing the surface. Notice that the coordinates range between -1.0 and 1.0. Your data will need to be converted to fit this range if they fall outside of these values.

### Additional Capabilities

At the bottom of the Graph3D window are two additional capabilities that can be applied to the graph: perspective and wire frame plotting. To change the display from a perspective projection to a parallel projection, click on the "Perspective" checkbox. The following two lines of code are executed when the checkbox is unselected:

```
View view = canvas.getView();
View.setProjectionPolicy(View.PARALLEL_PROJECTION);
```

The wire frame plot (see Figure 3) can only be accomplished with a trick – which is to set the four different color components of the Material class in Graph.java to be the same as the background color of the Canvas3D. The following four lines of code are executed in Graph.java when the "Fill On" checkbox is unselected:

```
material.setAmbientColor( backgroundColor );
material.setEmissiveColor( backgroundColor );
material.setSpecularColor( backgroundColor );
material.setDiffuseColor( backgroundColor );
```

The BranchGroup containing the lights (lightRoot) must also be detached from the hierarchy tree with lightRoot.detach(). Before this call is made, the following capability must be set in the makeLight() method in Graph3D:

```
lightRoot.setCapability(BranchGroup.ALLOW_DETACH);
```

### Installing Java3D

To run Graph3D, you'll have to have Java3D installed on your computer. Java3D can be downloaded at [java.sun.com/products/java-media/3D/downloads](http://java.sun.com/products/java-media/3D/downloads). If your computer is a PC, you may have to install OpenGL or DirectX. Most PCs automatically come with OpenGL or DirectX installed, but if yours doesn't, see [www.microsoft.com/downloads](http://www.microsoft.com/downloads). To run Java3D on Unix workstations see the workstation maker Web site for Java3D support.

### Running Graph3D

To compile the Graph3D application, type the following command in a DOS window from the directory containing the source code:

```
javac -classpath .; *.java
```

To run Graph3D as a standalone application in a DOS window, type the following command from the source code directory:

```
java -classpath .; Standalone
```

As with computer games, the quality of the Java3D display depends heavily on the quality of the graphics board installed in the computer. Low-end boards tend to leave out parts of the display like lines, surfaces, and text.

### Running from a Browser

Java3D can be run from both Netscape and Microsoft Internet Explorer (IE) web browsers. When installing the latest version of the Java JDK, the install process automatically points your web browser to the developer's kit. As long as you installed Java3D to this version of the JDK, all the work is done for you. To verify which version of the JDK your browser will use, go to the "Control Panel" on your PC and click on the Java menu.

To run Graph3D from one of these browsers, open the file AppletGraph3D.html (included in the source code) from your browser. Since Graph3D is expecting input from a data file on your local PC, you'll have to grant "read" file permission to the file plot3D.dat in the java.policy file. This is located in the "jdk\_install\_directory/jre/lib/security" directory. The syntax for granting permissions is:

```
grant codeBase "file:/C:/source_code/" {
  permission java.io.FilePermission "C:/source_code/plot3D.
  dat",read;
};
```

Granting permissions this way lets any applet running in the browser access your local file. So do this for demonstration purposes only. Realistically, a browser running the Graph3D applet won't use data from a file on the local computer. The user must change Graph3D appropriately to get the data from the server and format it.

### Summary

There are many fine commercial off-the-shelf (COTS) graphing packages available. While they may currently solve your particular graphing needs, writing your own graphing software gives you total control and lets you change it in any fashion desired. This will be beneficial in the long run since it will be totally portable and there will be no need for licenses, upgrades, or maintenance fees. Besides, the vendor may not incorporate any bug fixes or desired features not currently in a COTS package any time soon. Graph3D provides the core classes and technology for Java3D programming and can be used as a starting point for developing your own graphing application. As you've seen, Graph3D isn't a complex application and hopefully you'll consider using Java3D in your current or future projects. ☛





## See What's New at the 2006 JavaOne<sup>SM</sup> Conference

Come to the 11<sup>th</sup> Annual JavaOne<sup>SM</sup> conference and see where the Java<sup>TM</sup> platform is headed. More than 300 technical sessions and BOFs equip you with the knowledge you need for your current and future technological innovations.

Hear from industry experts, including Graham Hamilton and Bill Shannon of Sun Microsystems, as they discuss key directions for the next releases of the Java platform.

Plus, enhance your development skills in hundreds of expert-led sessions in five topics over four days:

- : Topic One :  
**Java Platform, Standard Edition (Java SE)**
- : Topic Two :  
**Java Platform, Enterprise Edition (Java EE)**
- : Topic Three :  
**Java Platform, Micro Edition (Java ME)**
- : Topic Four :  
**Tools**
- : Topic Five :  
**Cool Stuff**

**SAVE \$100!\*\***  
**REGISTER TODAY**  
at [java.sun.com/javaone/sf](http://java.sun.com/javaone/sf)

### GET CONNECTED

with the JavaOne Conference Event Connect tool.

As a Conference attendee, you can now connect with experts and fellow developers before the event. This effective online solution allows you to quickly schedule meetings with industry leaders and interact with other attendees.

Register for the Conference and get connected today.

### REFER YOUR FRIENDS

and receive an iPod nano music player!

One iPod nano music player per qualifying registrant. While supplies last.

*\*Content subject to change.*

*\*\*Offer not available on site.*

#### PLATINUM COSPONSORS



#### GOLD COSPONSORS



#### SILVER COSPONSORS



JavaOne<sup>SM</sup> Conference | May 16–19, 2006  
JavaOne<sup>SM</sup> Pavilion: May 16–18, 2006, Moscone Center, San Francisco, CA

**JavaOne<sup>SM</sup>**  
Sun's 2006 Worldwide Java Developer Conference

Copyright © 2006 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Java, the Java Coffee Cup logo, JavaOne, the JavaOne logo, Java Developer Conference, Java Community Process, JCP, 100% Pure Java, J2EE, J2ME, J2SE, Jini, Solaris, "Write Once, Run Anywhere," and all Java-based marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

# Using Self-Signed Certificates for Web Service Security

by Michael J. Remijan

## How to compete with trusted certificate authorities

One of the great things about the Java programming language is the Open Source community that provides great applications at little or no cost. An example of this is Apache Tomcat, which provides a solid Web server for development using servlet or JSP technology. Now that Web Service technology is maturing there's a potential for a whole scenario of applications to take advantage of a Swing feature-rich thin client on the front-end coupled to the data verification and business logic already developed in the Web or ejb tier. Such applications are only viable if they can be secure, however, security doesn't have to come at a great cost. The purpose of this article is to demonstrate how Web Service clients can use self-signed security certificates over the secure HTTPS protocol.

### The Problem with Using Self-Signed Certificates

HTTPS typically works seamlessly with the non-secure HTTP protocol and doesn't interrupt the user's experience. This is because SSL certificates are designed to be verified and signed by a trusted third party. Verisign is a popular certificate authority. If a Web application requires secure communication, you can pay Verisign to sign your SSL certificate. Once Verisign does that, users on your Web site can switch between HTTP and HTTPS without interruption because all major Web browsers trust certificates signed by Verisign. Verisign is not the only option for getting certificates signed. To save operating costs, or for personal use, you can self-sign your own certificate. However, self-signing your certificate will interrupt your Web site user's experience. Typically the Web browser will display a dialog box asking if you want to trust a certificate you signed.



Web browsers are nice because when they get a certificate signed by an unknown certificate authority there's an option to proceed. When developing Web Service clients for communication over HTTPS it's not so easy. When running Java code there's no dialog box asking about trusting a distrusted certificate authority. The JRE will throw an exception trying to connect over HTTPS to a Web site with a distrusted certificate:

Caused by: [sun.security.validator.ValidatorException: No trusted certificate found](#)

There's no way to catch this exception and continue. To get the Web Service to work with a self-signed certificate the JRE has to somehow trust you as a certificate authority.

### Solution Outline

To demonstrate a solution to this problem I'll do the following:

1. Generate and self-sign my own certificate
2. Configure Tomcat for SSL and make it use that certificate
3. Create an example Web Service to be called over HTTPS
4. Generate Web Service client code from WSDL
5. Demonstrate a client using a custom keystore solution

### Generating a Self-Signed Certificate

The JDK comes with a tool, `keytool`, that is used to manage SSL public/private keys. Keys are added and removed from a binary file on the file system. The default keystore file is `JAVA_HOME\jre\lib\security\cacerts`. This file contains the list of certificate authorities that the JRE will trust. A list of well-known trusted companies like Verisign is already in the keystore. To see this list, execute with password "changeit":

```
D:\>keytool -list -rfc -keystore JAVA_HOME\
jre\lib\security\cacerts
```

The `keytool` application can be used to edit this file. However, just in case something goes wrong it's better to create a new file. If `keytool` isn't told which file to use it creates `HOME/.keystore` by default.

To generate your own self-signed certificate execute:

```
D:\>keytool.exe -genkey -alias Tomcat -key-
alg RSA -storepass bigsecret -keypass big-
secret -dname "cn=localhost"
```

After executing this command there will be a `.keystore` file in your HOME directory. Here's what the switches mean.

- **genkey:** Tells the `keytool` application to generate new public/private key pair.
- **alias:** The name used to refer to the keys. Remember, the `.keystore` file can contain many keys.
- **keyalg:** Generates public/private keys using the RSA algorithm.
- **storepass:** What password is needed to access the `.keystore` file.
- **keypass:** What password is needed to manage the keys.



**Michael J. Remijan** is a senior Java consultant at Technology Partners in St. Louis, MO. He designs and develops mission-critical J2EE applications. Michael has a BS in mathematics/computer science from the University of Illinois and an MBA in technology management from the University of Phoenix.  
mjremijan@yahoo.com

- **dname:** This value is very important. I used “localhost” because this example is designed to run locally. If a Web application is registered as <http://www.myserver.com> then this value must be [www.myserver.com](http://www.myserver.com). If the names don't match the certificate will automatically be rejected.

Once the keytool application creates a new public/private key pair it automatically self-signs the key. You have just generated your own self-signed certificate, which can be used for HTTPS communications. You only need to extract the self-signed public key. I'll show how to do this later.

## Configuring Tomcat for SSL

Now you have to configure Tomcat to use your self-signed certificate. I used Tomcat 5.0.30. Edit the `TOMCAT/conf/server.xml` file. Search the file for “8443” and uncomment the `<Connector.../>` bound to that port. Then you'll have to add the following property to the `<Connector.../>`:

```
keyStorePass="bigsecret"
```

When the JRE starts, it will automatically find the `HOME/.keystore` file and Tomcat will try to access it using the password “bigsecret.” When Tomcat starts there should be output to the console that looks similar to:

```
Feb 4, 2006 3:11:23 PM org.apache.coyote.  
http11.Http11Protocol start  
INFO: Starting Coyote HTTP/1.1 on http-8443
```

This means the `<Connector.../>` successfully read the `.keystore` file and you can now do secure HTTPS connections over the 8443 port. Open a Web browser and try <https://localhost:8443/>. Because the certificate is self-signed the Web browser will display a dialog box asking about trusting the connection. If accepted, all communications will be secure over HTTPS.

## Creating the Web Service

I'm going to use the Apache Axis project to create a very simple Web Service. The Web Service will simulate checking for new e-mail messages. A Web Service client passes a token uniquely identifying a user. The Web Service returns a list of new e-mail messages (see Listing 1).

To get the Web Service deployed, follow these steps:

1. Cut and paste the code from Listing 1 into a file named `Email.jws` in Webapp's root directory.
2. Edit the `Web.xml` file, adding the Axis servlet and a `*.jws` mapping (Listing 2).
3. Put the Axis jar files in `WEB-INF/lib`. See References at the end of this article for the Axis project URL.

After deploying this article's accompanying WAR file (and configuring Tomcat for SSL), the Web Service is accessible securely over HTTPS at the following URL:

<https://localhost:8443/JDJAArticleWebService/Email.jws>

## Using WSDL2Java

The Axis project provides a tool named `WSDL2Java` that takes a Web Service WSDL and automatically create the Java source code needed to use the Web Service. See Listing 3 for the command line used to generate code for the `Email.jws` Web Service.

Notice the URL in Listing 3 used to access the WSDL. It's the non-secure HTTP protocol over port 8080. Why not use HTTPS over port 8443? Because of the self-signed certificate, the `WSDL2Java` tool will encounter the same exact certificate problem this article is trying to provide a solution for. So for now the non-secure protocol must be used. This means the generated code must be altered slightly replacing “http” and “8080” references with “https” and “8443.” This article's accompanying client zip file contains the altered code.

## Client with a Custom Keystore

The JRE's default keystore is `JAVA_HOME\jre\lib\security\cacerts`. Java applications will throw an exception whenever they are presented with your self-signed certificate because your certificate isn't in this keystore. Therefore, when developing a client there are two options. The first option is to put your self-signed certificate into the JRE's default keystore. Although this would work it's not a very good solution because customization is required on every client machine. The second option is to generate a custom keystore, put your self-signed certificate into it,

and distribute the custom keystore as part of your application (typically inside a jar file).

To create a custom keystore for your client the following has to be done:

1. Export the self-signed public key from `HOME/.keystore`.
2. Import the self-signed public key into a new keystore for you client.

To export the self-signed public key from `HOME/.keystore` execute the following:

```
D:\>keytool.exe -export -rfc -alias Tomcat  
-file Tomcat.cer -storepass bigsecret -key-  
pass bigsecret
```

Now create the custom keystore for the client by importing `Tomcat.cer`:

```
D:>keytool.exe -import -noprompt -trust-  
cacerts -alias Tomcat -file Tomcat.cer  
-keystore CustomKeystore -storepass lit-  
tlesecret
```

Using the switch “-keystore `CustomKeystore`” will create a new keystore file called “`CustomKeystore`” in the present working directory. You'll find the `CustomKeystore` file in the `/classpath/resources/keystore` directory of this article's client zip file. Replace this one with the file just generated.

Now all that's left to do is to create a client that uses this custom keystore. I'll demonstrate two ways to do this.

The first is to use the Java system properties `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` to point to the `CustomKeystore` file and provide the password to access it. My example Web Service client in the `jdj.wsclient.truststore` package takes this approach (see Listing 4). The `main()` method sets the system properties then creates the objects to use the Web Service. When the JRE needs to access a keystore it looks for the “`classpath/resources/keystore/CustomKeystore`” file on the file system. Although this is a simple solution it's problematic because the keystore file must be on the file system and the client code must know where to look for it.

The second is a more portable solution that keeps resources inside the jar file and avoids the file system issues. The client code is responsible for



reading the CustomKeystore file and somehow using it to create a secure connection to the server. My example Web Service client in the `jdj.wsclient.socketfactory` package takes this approach (see Listing 5). Listing 5 shows how to read the CustomKeystore file as a resource and use it to create a `javax.net.ssl.SSLSocketFactory`. Configuring the Axis pluggable architecture, the `MySocketFactory` class can be then used to create secure Socket objects from this factory.

## Conclusion

This article started with a simple problem: I wanted to secure Web Service communications over HTTPS using my own self-signed certificates. By default, the JRE will reject my application's self-signed certificate because I am not a trusted certificate authority. To get secure communications to work I had to get the Web Service client JRE to trust my self-signed certificate. To achieve this, I used the `keytool` application and

generated a new public/private key pair, extracted the self-signed public key, and then created a new keystore and imported this self-signed certificate. I then created a totally self-contained Web Service client that doesn't require any client-side configuration. ☛

## References

- <http://tomcat.apache.org/tomcat-5.0-doc/ssl-howto.html>
- <http://ws.apache.org/axis/>

### Listing 1

```
import java.util.*;

public class Email {

    public List
    getNewMessages(String id)
    {
        List l = new ArrayList(3);
        l.add("1");
        l.add("2");
        l.add("3");
        return l;
    }
}
```

### Listing 2

```
<servlet>
  <servlet-name>
    AxisServlet
  </servlet-name>
  <display-name>
    Apache-Axis Servlet
  </display-name>
  <servlet-class>
    org.apache.axis.transport.http.AxisServlet
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>
    AxisServlet
  </servlet-name>
  <url-pattern>
    *.jws
  </url-pattern>
</servlet-mapping>
```

### Listing 3

```
java
-classpath
.;axis.jar;log4j-1.2.8.jar
;commons-logging-1.0.4.jar
;commons-discovery-0.2.jar
;jaxrpc.jar;saaj.jar
;wsdl4j-1.5.1.jar
org.apache.axis.wsdl.WSDL2Java
-p jdj.wsclient.shared
http://localhost:8080/JDJArticle/Email.jws?wsdl
```

### Listing 4

```
public static
void main(String[] args)
throws Exception
{
    System.setProperty(
        "javax.net.ssl.trustStore",
        "classpath/resources/keystore/CustomKeystore");

    System.setProperty(
```

```
"javax.net.ssl.trustStorePassword",
"littlesecret");

EmailServiceLocator wsl =
    new EmailServiceLocator();

Email_PortType ews =
    wsl.getEmail();

Object [] objects =
    ews.getNewMessages("12345");

out("Msg Count: " + objects.length);
}
```

### Listing 5

```
public MySocketFactory(Hashtable table)
throws Exception
{
    out("Created!");
    KeyStore ks =
        KeyStore.getInstance(
            KeyStore.getDefaultType()
        );

    char [] password =
        "littlesecret".toCharArray();

    String keystore =
        "/resources/keystore/CustomKeystore";

    Class tclass =
        this.getClass();

    InputStream is =
        tclass.getResourceAsStream(
            keystore
        );

    ks.load(is, password);

    KeyManagerFactory kmf =
        KeyManagerFactory.getInstance("SunX509");

    kmf.init(ks,password);

    TrustManagerFactory tmf =
        TrustManagerFactory.getInstance("SunX509");

    tmf.init(ks);

    SSLContext context =
        SSLContext.getInstance("SSL");

    context.init(
        kmf.getKeyManagers(),
        tmf.getTrustManagers(),
        new SecureRandom()
    );

    factory =
        context.getSocketFactory();
}
```

Advertiser	URL	Phone	Page
AJAX Seminar	<a href="http://www.ajaxseminar.com">www.ajaxseminar.com</a>	201-802-3022	48-49
Altova	<a href="http://www.altova.com">www.altova.com</a>	978-816-1600	4, 21
Backbase	<a href="http://www.backbase.com">www.backbase.com</a>	866-800-8996	31
Enerjy	<a href="http://www.enerjy.com/visible">www.enerjy.com/visible</a>	866-598-9876	29
Fiorano	<a href="http://www.fiorano.com/downloadsoa">www.fiorano.com/downloadsoa</a>	800-663-3621	19
Instantiations	<a href="http://www.instantiations.com/rcpdeveloper">www.instantiations.com/rcpdeveloper</a>	800-808-3737	17
InterSystems	<a href="http://www.intersystems.com/javaone2006p">www.intersystems.com/javaone2006p</a>	617-621-0600	7
IT Solutions Guide	<a href="http://www.itsolutions.sys-con.com">www.itsolutions.sys-con.com</a>	888-303-5282	57
JadeLiquid Software	<a href="http://www.webrenderere.com">www.webrenderere.com</a>	+61-3-6226-6274	9
JavaOne Conference 2006	<a href="http://java.sun.com/javaone/sf">java.sun.com/javaone/sf</a>		53
Northwoods Software Corp.	<a href="http://www.nwoods.com">www.nwoods.com</a>	800-434-9820	45
ODTUG	<a href="http://www.odtug.com">www.odtug.com</a>		43
OpenAccess Software, Inc.	<a href="http://www.openaccesssoftware.com/jdj">www.openaccesssoftware.com/jdj</a>	888-805-ODBC	23
OPNET Technologies, Inc.	<a href="http://www.opnet.com/pinpoint">www.opnet.com/pinpoint</a>	240-497-3000	37
Parasoft Corporation	<a href="http://www.parasoft.com/jdjmagazine">www.parasoft.com/jdjmagazine</a>	888-305-0041	Cover IV
Parasoft Corporation	<a href="http://www.parasoft.com/wsjmagazine">www.parasoft.com/wsjmagazine</a>	888-305-0041	47
Perforce	<a href="http://www.perforce.com">www.perforce.com</a>	510-864-7400	11
Quest Software	<a href="http://www.quest.com/javacode">www.quest.com/javacode</a>	949-754-8000	Cover II
Roaring Penguin	<a href="http://www.roaringpenguin.com">www.roaringpenguin.com</a>	613-231-6599	34, 35
SOA/EOS Conference	<a href="http://www.soaeosconference.sys-con.com">www.soaeosconference.sys-con.com</a>	201-802-3022	60-61
Software FX	<a href="http://www.softwarefx.com">www.softwarefx.com</a>	800-392-4278	Cover III
SUN Microsystems	<a href="http://www.netbeans.org">www.netbeans.org</a>		13
TIBCO Software Inc.	<a href="http://www.tibco.com/mk/gi">www.tibco.com/mk/gi</a>	800-420-8450	15

**General Conditions:** The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of *Java Developer's Journal*. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in *Java Developer's Journal*. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc.

This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

# 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.

# Meet the *JDJ* Editors

**W**e thought it was time that the readers of *JDJ* had a chance to meet the editors, those individuals behind the scenes who work tirelessly to bring you the best articles about Java in particular and *i*-Technology in general.

Over the next few issues, the editors will provide a brief glimpse into their daily lives, their likes and dislikes, why they like to write, and more.



**Yakov Fain**  
Java EE Editor

**Q** What's your primary job?

**YF:** I'm a Java consultant, working for Wall Street companies.

**Q** What's your typical day like?

**YF:** I wake up at 5 a.m. and spend an hour browsing the Internet and writing, then another hour of the same activity on the bus while commuting to work. The next 8–9 hours are spent working for whoever my current client is doing Java code review, writing technical proposals, coding, performance improvements, evaluating third-party tools to make sure that my client does not purchase fluff. After that, another hour of intimate relations with my laptop on the way home.

**Q** Why do you write for *JDJ*?

**YF:** I like to write and, as with every author, I want to have lots of readers.

*JDJ* is one of the handful of Java-related publications with a really large number of subscribers. Also, being a published author helps me find consulting jobs. Actually, on one of the job interviews I was asked, "Don't you think that your writing will prevent you from staying focused on our project?" Luckily, most managers are smarter than this one.

**Q** Do you blog?

**YF:** Yes, I do at <http://yakovfain.javadevelopersjournal.com>. This is a place where I can quickly record my thoughts. My blog also helps me to promote my books or Java training, which I run once in a while.

**Q** What do you like about Java?

**YF:** This is a living language with a huge following. It has lots of tools, frameworks, and components, and competition forces vendors to make them better and better.

**Q** What don't you like about Java?

**YF:** It has too many tools, frameworks, and components. You must learn too many things to stay afloat. The days when knowledge of a one front-end tool and a DBMS would land you a job are gone.



**Q** What would be a perfect job for you?

**YF:** Writing the courseware and traveling around the world teaching programming languages.

**Q** What's the most exciting project you're working on now?

**YF:** I'm working on a book called *Rich Internet Applications with Adobe Flex and Java* ([www.theriabook.com](http://www.theriabook.com)). Flex is a nicely designed technology, and this duo will be hot next year.

**Q** What's your hobby?

**YF:** Last year we went fishing. We hired a captain with a boat and the captain said, "It was a great season; I've been working for 42 days without taking even one day off. Tomorrow is my first day off." When we asked him how he was planning to spend this day, he answered that he'd be going fishing. I have to admit, programming is my main hobby. I also love to travel and gourmet food.

**Q** Do you have children? If yes, would you like them to be computer programmers?

**YF:** My son Yuri is graduating from college with a degree in classical animation. He was always a good artist and we let him study what he's good at. By the way, anyone have a job for a young animator? My son David is a sixth grader. He's more into computers and I'll be happy if he decides to become a programmer, but it's going to be his call.

“

*JDJ* is one of the handful of Java-related publications with a really big subscriber list”

—Yakov Fain





## The Java community is doing some really cool stuff in the enterprise space (Spring EJB3, 16 Web frameworks :-)"

—Bill Dudney

**Q** *What are the three most important things you've learned during your life in IT?*

- YF:** (1) Be a team player. Never act like a prima donna.  
 (2) In large corporations, political reasons are the main factor when making technical decisions.  
 (3) Be loyal to your employer, but always keep your skills and your résumé up to date. The blue sky can turn grey within one day, and then things get ugly.

**Q** *What do you think about IT outsourcing?*

- YF:** IT outsourcing stinks for three reasons:  
 (1) it hits the weakest, who are mid-age women programmers;  
 (2) they don't feel secure, hence they don't recommend IT as a career for their children;  
 (3) half of the enterprise managers don't know how to handle offshore programmers, hence many projects fail.

One great thing about outsourcing is that it makes the IT market more competitive. Just be a little better than a mediocre programmer and you'll always have a job.



**Bill Dudney**  
Contributing Editor

**Q** *What's your primary job?*

**BD:** Herding cats, oh wait, I mean leading a team of amazingly great folks at Virtuas. We help clients adopt open source and my primary role in that is focused around Hibernate and MyFaces.

**Q** *What's your typical day like?*

**BD:** A typical day is spent writing (code, articles, training material, documentation, RFPs, etc.), speaking, meetings, phone calls, and more.

**Q** *Why do you write for JDJ?*

**BD:** It's fun to share the knowledge that I've gained in the trenches. I'd rather spend two hours writing up what has finally sunk in after beating my head against the monitor for 16 hours than having someone else get a bruised forehead. Also I have the wonderful privilege of being able to do a lot of "cutting-edge" stuff and so that is always fun to write about because not everyone has that kind of time.

**Q** *Do you blog?*

**BD:** Yes, at <http://bill.dudney.net/roller/page/bill>.

**Q** *What do you like about Java?*

**BD:** The Java community is doing some really cool stuff in the enterprise space (Spring, EJB3, 16 Web frameworks :-). It's fun to be part of a huge group of such talented folks.

**Q** *What don't you like about Java?*

**BD:** The impl of Java is not open source, so I'm stuck waiting on Apple to release the next version of Java for my PowerBook...

**Q** *What would be a perfect job for you?*

**BD:** The one I'm doing now :-)

**Q** *What's the most exciting project you are working on now?*

**JB:** Cayenne and the MyFaces JSF 1.2 stuff.

**Q** *What's your hobby?*

**BD:** Creating DVDs of the family videos. I shoot video of the kids all year, then edit them into masterpieces with the Apple video editing suite and burn them onto DVDs to the delight of grandparents on both sides of the family (at least that's what they tell me :)

**Q** *Do you have children? If yes, would you like them to be computer programmers?*

**BD:** Sure, my oldest (Andrew, age 10) is doing NQC for his Lego Mindstorm robots. None of the others have expressed interest yet, but my 8-year-old is interested in Mathematica, so he might become a hacker yet :-)

**Q** *What are the three most important things you've learned during your life in IT?*

- BD:** (1) If you don't know what to build, you will build something but it won't be what the user wants.  
 (2) Users have no idea how to communicate with developers and developers are more or less clueless about business – get something done now and show it to the users. If they are happy with it, build more; if they're not, fix it right away before moving on to the next feature.  
 (3) Community (i.e., people) is so much more important than code. With a good community you can fix anything, but without a good community you don't stand a chance. This applies in commercial IT development as much as it does in open source.

**Q** *What have we not asked?*

**BD:** Can't possibly fit my soap box into 450 words or less :-). But the bottom line is that we need open source to have an open and diverse developer community. OS projects that are dominated by one company run the risk of being annihilated by acquisition or another form of hostile action. I have no axe to grind about any particular community or company, it's just a trend that I've seen in the open source commercialization space that troubles me. ☹

BONUS TRACK ADDED! (FREE \$1,295 VALUE!)  
"Real-World AJAX" 2-Day Seminar

# SOA WebServices, Enterprise OpenSource, + Real-World AJAX!

## The #1 i-Technology Educational and

Register at... [www.SOAEOSConference.sys-con.com](http://www.SOAEOSConference.sys-con.com)

The **10th International SOA Web Services Edge Conference**, colocated with the **First Annual Enterprise Open Source Conference**, will take place on **June 5-6, 2006** at the historical **Roosevelt Hotel in New York City**. SOA Web Services Edge and Enterprise Open Source Conference together deliver the #1 i-technology educational and networking opportunity of the year.

This year, more than 100 distinguished conference faculty members will present **96 cutting-edge SOA, Web Services and Enterprise Open Source topics** in educational classes, panels, and keynotes in six simultaneous tracks for two information-packed days, plus a bonus track, a **"Real-World AJAX"** two-day seminar, a \$1,295 value!

The conference program does not merely present a comprehensive view of all the development and management aspects of integrating a **SOA strategy** and an **Open Source philosophy** into your enterprise, its organizing principle is that delegates will go away from the intense two-day program replete with why-to and how-to knowledge delivered first-hand by industry experts.

According to analyst firm Gartner Group, **by 2008 more than 60 percent of enterprises will use SOA** as the guiding principle when creating mission-critical applications and processes. **"Businesses that ignore the potential of SOA will find themselves outpaced by rivals who improve their agility and transform themselves into new kinds of enterprises,"** says Gartner analyst Yafim Natis.

Meanwhile, the **Open Source revolution** has spread into traditional areas of enterprise IT, with all major vendors and a global group of **Open Source entrepreneurs** developing the technology and delivering the solutions that are changing the way the world does business and the way governments serve their citizens.

SPONSOR BY:



Receive **FREE**  
WebCast Archives  
of all Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all 96 conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.



► This on-demand archives set  
is sold separately for \$995

## Receive 8 Days Worth of Education F



**Jeffrey Barr  
Amazon**

As Web Services Evangelist for Amazon.com, Jeff Barr focuses on creating developer awareness for the Amazon software platform. He has a longstanding interest in Web services and programmatic information interchange. Jeff has held development and management positions at KnowNow, eBay, Akopia, and Microsoft, and was a co-founder of Visix Software. Jeff's interests include collecting and organizing news feeds using his site, [www.syndic8.com](http://www.syndic8.com). He holds a Bachelor's Degree in Computer Science from the American University and has done graduate work in Computer Science at the George Washington University.



**Israel Hilerio  
Microsoft**

Israel Hilerio is a program manager at Microsoft in the Windows Workflow Foundation team. He has 15+ years of development experience doing business applications and has a PhD in Computer Science.



**Adam Kolawa  
Parasoft**

Adam Kolawa, Parasoft co-founder and CEO, is considered to be a visionary in his field. In 1983, he came to the United States from Poland to pursue his Ph.D. In 1987, he and a group of fellow graduate students founded Parasoft to create value-added products that could significantly improve the software development process. Kolawa's years of experience with various software development processes has resulted in his unique insight into the high-tech industry and the uncanny ability to successfully identify technology trends. As a result, he has orchestrated the development of numerous successful commercial software products to meet growing industry needs to improve software quality.



**Jason Levitt  
Yahoo!**

Jason Levitt, Technical Evangelist on creating Flash-based Yahoo! Maps applications.



**Duane Nickull  
Adobe**

As senior standards strategist for Adobe Systems, Duane Nickull is responsible for managing Adobe's participation in OASIS and UN/CEFACT, as well as ensuring that Adobe's enterprise solutions support emerging XML standards. Previously Mr. Nickull co-founded Yellow Dragon Software Corporation, a privately held developer of XML messaging and metadata management software, recently acquired by Adobe. Mr. Nickull currently serves as a vice chair of the United Nations Centre for Facilitation of Commerce and Trade (UN/CEFACT) where he oversees the United Nations Electronic Business strategy and architecture.



**Bob Pasker  
Azul**

Bob Pasker is deputy CTO with Azul Systems. He has been designing and developing networking, communications, transaction processing, and database products for 25 years. As one of the founders of WebLogic, the first independent Java company (acquired by BEA Systems in 1998), he was the chief architect of the WebLogic Application Server. Bob has provided technical leadership and management for numerous award-winning technologies, including the Tribelink series of routers and remote access devices, and the TMX transaction processing system. Bob graduated magna cum laude and Phi Beta Kappa from San Francisco State University and holds a Masters degree from Brown University.



**Brian Behlendorf  
CollabNet**

Brian Behlendorf founded CollabNet, with O'Reilly & Associates, in July 1999. The company provides tools and services based on open source methods. Before launching CollabNet, Behlendorf was co-founder and CTO of Organic Online, a Web design and engineering consultancy located in San Francisco. During his five years at Organic, Behlendorf helped create Internet strategies for dozens of Fortune 500 companies. During that time, he co-founded and contributed heavily to the Apache Web Server Project, co-founded and supported the VRML (Virtual Reality Modeling Language) effort, and assisted several IETF working groups, particularly the HTTP standardization effort.



**Marc Fleury  
JBoss**

Born in Paris in 1968, Marc Fleury got his Ph.D in physics from the Ecole Polytechnique in Paris. He started in Sales at Sun Microsystems France and then moved to the US where he worked on early java enablement of SAP at SAP Labs. Marc started the JBoss project in 1999. An ex-Lieutenant in the paratroopers, Marc holds a degree in Mathematics from the Ecole Polytechnique, a master in Theoretical Physics from the Ecole Normale ULM and was a visiting scientist at MIT during his thesis. Marc's research interest focuses on aspect oriented middleware.



**Andy Astor  
EnterpriseDB**

Andy is President and CEO EnterpriseDB, the world's leading enterprise-class, open source database company. Previously, Andy was vice president webMethods, leading the company's open source, standards, and Web services agendas. Andy was elected twice to the Board of Directors of the Web Services Interoperability Organization (WS-I), and led WS-I's marketing efforts. Prior to joining webMethods, Andy was vice president at D&B, where he led worldwide development of all on-line products. His work at D&B included the development and launch of one of the earliest commercial Web services.



**Mike Milinkovich  
Eclipse.org**

Mike Milinkovich has held key management positions at Oracle, WebGain, The Object People, and Object Technology International Inc. (which subsequently became a wholly-owned subsidiary of IBM), assuming responsibility for development, product management, marketing, strategic planning, finance and business development. Mike earned his MS degree in information and systems sciences and a bachelor of commerce degree from Carleton University in Ottawa, Canada.



**Peter Yared  
ActiveGrid**

Peter Yared is the founder and CEO of ActiveGrid. Most recently, he was CTO of Sun Microsystems's Liberty Network Identity initiative. Mr. Yared was also CTO of Sun Microsystems Application Server Division. Before its acquisition by Sun, Mr. Yared served as CTO of NetDynamics, which pioneered the then-leading J2EE application server. Earlier, Mr. Yared was founder and CEO of JRad Technologies, an enterprise Java company acquired by NetDynamics. Additionally, Mr. Yared was Chief Architect of client/server products at object-oriented tool maker Prograph International and the architect of several mission-critical systems deployed by U.S. government agencies and the GED Testing Service.



**David Temkin  
Laszlo**

David Temkin is Chief Technology Officer of Laszlo Systems, Inc. In this role, he has positioned the company to become the next technology standard for rich Internet applications. Under his direction, Laszlo developed its patent-pending open-source product suite and extended operations to both coasts of the United States. Before founding Laszlo, Temkin was senior director of engineering at Excite@Home where he led a team of 55 engineers, designers and technical writers responsible for developing the company's consumer software. Prior to Excite@Home, Temkin was an engineering manager in the Newton division at Apple Computer and developed enterprise software at EDS.



**Kevin Hakman  
TIBCO**

Kevin Hakman is Co-founder, TIBCO General Interface, TIBCO Software Inc. Prior to TIBCO General Interface, he was the co-founder of Versant Inc. a leading provider of enterprise client technology. Prior to Versant, he founded a series of successful emerging Internet technology and e-commerce ventures. He has also written for eBusiness Journal and HotWired.



**Coach Wei  
Nexaweb**

Coach Wei currently serves as CTO for Nexaweb, which develops the leading XML-based rich client technology platform for building and deploying Enterprise Internet Applications. Previously, he played a key role at EMC Corporation in the development of a new generation of storage network management software. Coach is a graduate from MIT, holds several patents, and is an industry advocate for the proliferation of open standards.



**Luis Derechin  
JackBe**

Luis Derechin is CEO and Co-Founder of JackBe. Mr. Derechin has over 12 years of entrepreneurial and management experience. He has been part of the founding team of successful startups, including a catalogue retail company that achieved \$15M in sales.



**Jouk Pleiter  
Backbase**

Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful Software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market.

# Networking Opportunity of the Year!

**SOA 10th International  
WebServices  
Edge** conference  
**ENTERPRISE >  
OPENSOURCE  
CONFERENCE**

**REAL-WORLD  
AJAX  
SEMINAR**

**Discounted Price**  
Incl. Golden Pass to Both Conferences\* **\$1,495**

**Conference Price** (On June 5-6, 2006)  
Incl. Golden Pass to Both Conferences\* **\$1,695**

\*BONUS: "Real-World AJAX" Two-Day Seminar!

**Attention Sponsors:**

A Forum will be available to display leading Web services, OpenSource, and AJAX products, services, and solutions.



**JUNE 5-6, 2006**  
Roosevelt Hotel / New York, NY

**For Sponsorship Information > Call 201-802-3020**

**TOPICS INCLUDE...**

**SOA Web Services Edge:**

- > Transitioning Successfully to SOA
- > ebXML
- > Orchestration
- > The Business Case for SOA
- > Interop & Standards
- > Web Services Management
- > Messaging Buses and SOA
- > SOBAs (Service-Oriented Business Apps)
- > Delivering ROI with SOA
- > Java & XML Web Services
- > Security
- > Systems Integration
- > Sarbanes-Oxley
- > Business Process Management

**Enterprise Open Source:**

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

**AJAX User Experience:**

- > Rich Internet Applications
- > Flash-AJAX Integration
- > Developing Interfaces
- > AJAX on Rails
- > Best Practices
- > Enterprise AJAX
- > Technology Demos
- > Lessons from the Frontline
- > Bringing Desktop Apps to the Web
- > AJAX Power Panel
- > and more...

**sys-con** EVENTS [www.EVENTS.sys-con.com](http://www.EVENTS.sys-con.com)

# Packed into 2 Information-Filled Days!





Onno Kluyt

# 2006 JavaOne JSR Itineraries

It's JavaOne show time again. The Java Community Process (JCP) Program and its members have a lot to share from the latest Java specification (JSR) accomplishments showcased in a diversity of forms at the conference, including technical sessions (TS), birds-of-a-feather meetings (BOF), industry panels, training sessions, round tables, and community events. Let me give you a mini tour of some of the JSRs on the conference agenda this year.

I'll start with the Technical General Session featuring Graham Hamilton and Bill Shannon, Sun Microsystems. Their talk will present key directions of the Java Platform, Standard Edition 6 (Java SE 6), and the Java Platform, Enterprise Edition 5 (Java EE 5), future releases both developed based on two Java platform specifications (JSR) built through the JCP Program: JSR 270 and JSR 244. You'll want to hear from the architects what's under the hood today and what the future holds. If you want to dive into certain aspects of the specs, check out the individual technical sessions based on them included in the Content Catalog Viewer of the Conference (accessible from <http://java.sun.com/javaone/sf/sessions.jsp>).

Java API for XML-Based Web Services (JAX-WS) 2.0, JSR 224, is showcased in TS 1194. If you're a developer of client applications, you probably already put it on your conference schedule. Even if your interest is broader, you'll want to check out this session presented by the spec leads of the JSR, Rajiv Mordani and Roberto Chinnici, both from Sun. The major focus of this standard is ease-of-development in order to allow the technology to be used by a wide circle of developers and simplify their tasks. The specification extends Java APIs for XML-based RPC in a number of different areas including alignment with JSR 181, Web Services Metadata for the Java Platform. Attend the TS and you'll find out how the spec strongly aligns with Java Architecture for XML Binding (JAXB) 2.0, to which it delegates all data binding-related tasks, and how it supports new versions of external standards from organizations such as W3C and WS-I.

More about JAXB 2.0, JSR 222, can be learned in this JSR's own technical session, TS 1607. Driven by Sun, this standard is the

next version of JAXB, the Java Architecture for XML Binding. It brings additional functionality while retaining ease of development, and provides support for W3C XML Schema features including frequently requested features such as type and element substitution.

Another JSR-based session is Version 2.0 of the Portlet Specification, JSR 286, headed by IBM. It's presented at TS 3627 and showcases the functionality that will be added to the new Portlet Specifications. This API is based on the version 1.0 defined in JSR 168 and will be binary-compatible with version 1.0. Mark this session on your calendar to find out about the features and requirements that will be addressed in this JSR.

If you want to find out what's being planned for Java Archives (JARs) used as distribution and execution format for Java applications, include JSR 277, Java Module System – BOF 0684, in your JavaOne itinerary. The specification developed by Sun sets out to define a distribution format and a repository for collections of Java code and related resources as well as discovery, loading, and integrity mechanisms at runtime.

The spec leads will present the thinking and work around uniform deployment models for Java applications and extensions, resolving versioning conflicts and namespace collision among Java extensions in a JRE, and sharing Java extensions among different JREs on the same system.

Talked about widely and capturing the interest of all developers, JSR 220, Enterprise JavaBeans 3.0, has its own technical session, TS 1887. Co-led by Sun and Oracle, the standard improves the EJB architecture, making it easier for a wider range of developers to use it. Attend the technical session to discover the exciting improvements the API incorporates.

Another Sun-led JSR, 269, Pluggable Annotation Preprocessing API, is the topic of BOF 0606. The spec lead and members of the JSR Expert Group get together to offer an update on adding standard annotations processing to JDK 6. Full benefits of annotations is expected to be realized at the end of this standardization work.

IBM and BEA, the drivers of JSR 235, Service Data Objects (SDO), partner at TS-3676 to talk about how developers will be able to simplify data access and representa-

tion in service-oriented software by replacing data access models with a uniform abstraction for creating, retrieving, updating, and deleting business data used by service implementations.

Two spec leads from Nokia team up to present at BOF 2810, the recently finalized JSR 256, Mobile Sensor API. The standard defines basic sensor functionality for mobile devices and extends the usability and choice of sensors for Java ME applications. If you want to learn how new applications are enabled by the use of sensors, attend this hands-on BOF.

Another partnership, this time between Nokia and Vodafone spec leads, brings to life JSR 248, Mobile Service Architecture featured in TS-4936. It's a tutorial you don't want to miss if you desire to have a sneak-peak at the rich functionality designed to create a highly predictive Java platform for mobile devices.

JSR 232, Mobile Operational Management, led by Motorola and Nokia, is the topic of TS-3757. The session will offer an introduction to the spec under development through the JCP Program and will explain the benefits this standard sets out to pass on to developers.

Another JSR led by Nokia, JSR 257, Contactless Communication API, is presented at TS-3789. Close to final development stage at the time of writing, this specification defines a standardized way to utilize contactless communication in Java ME applications based, for example, on RFID, NFC (Near Field Communication), or bar codes. Developers interested in code examples and demos of how JSR 257 Contactless Communication API can be used should attend.

Nokia and Motorola, co-spec leads of JSR 272, Mobile Broadcast Service API for Hand-held Terminals, will present at TS 4693 the features of this set of APIs. In early development stages under the JCP, the spec will allow designing device-independent broadcast applications. If you want to get an overview of the technology and the APIs along with code samples from the experts, this is a tech session you shouldn't miss.

This is just a sampling of JSR-based technology events at the 2006 JavaOne Conference rather than a complete overview. For a complete search of JSR-based sessions, go to <http://java.sun.com/javaone/sf/sessions.jsp>.

Onno Kluyt is the director of the JCP Program Management Office, Sun Microsystems.  
[onno@jcp.org](mailto:onno@jcp.org)

**SoftwareFX**

Bringing Your Data, Business & Strategy Into Focus.

**New!**  
**version 6.2**  
**Now Includes Maps!**

# All New Flavors... Same Great Taste of Java!

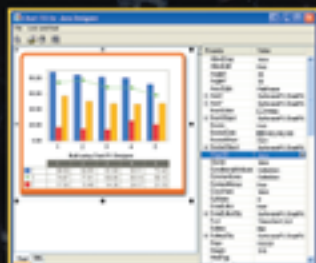


Chart FX for Java Designer

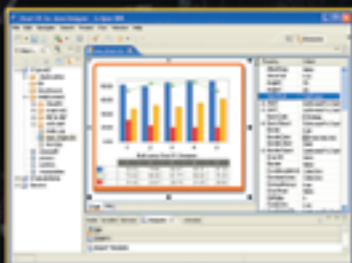


Chart FX for Java - Eclipse

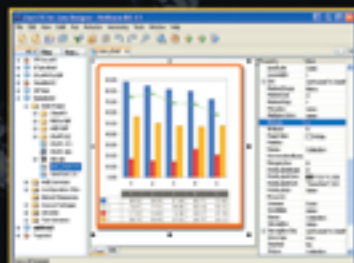


Chart FX for Java - NetBeans

## Chart FX for Java 6.2

Now offers seamless integration within Eclipse and NetBeans, Chart FX for Java 6.2 brings New features include Smart & dynamic axis labeling, conditional marker and axis attributes, extended URL linking, multiple chart panes and robust annotation objects. One of the most dynamic new features is the inclusion of the Chart FX Maps Extension as a permanent fixture within Chart FX for Java. ➤



  
**Chart FX**

US: (800) 392-4278 • UK: +44 (0) 8700 272 200 • Check our website for a Reseller near you!

[www.softwarefx.com](http://www.softwarefx.com)

©2006 Software FX. All rights reserved. Chart FX is a registered trademark of Software FX, Inc. All other brands are owned by their respective owners.

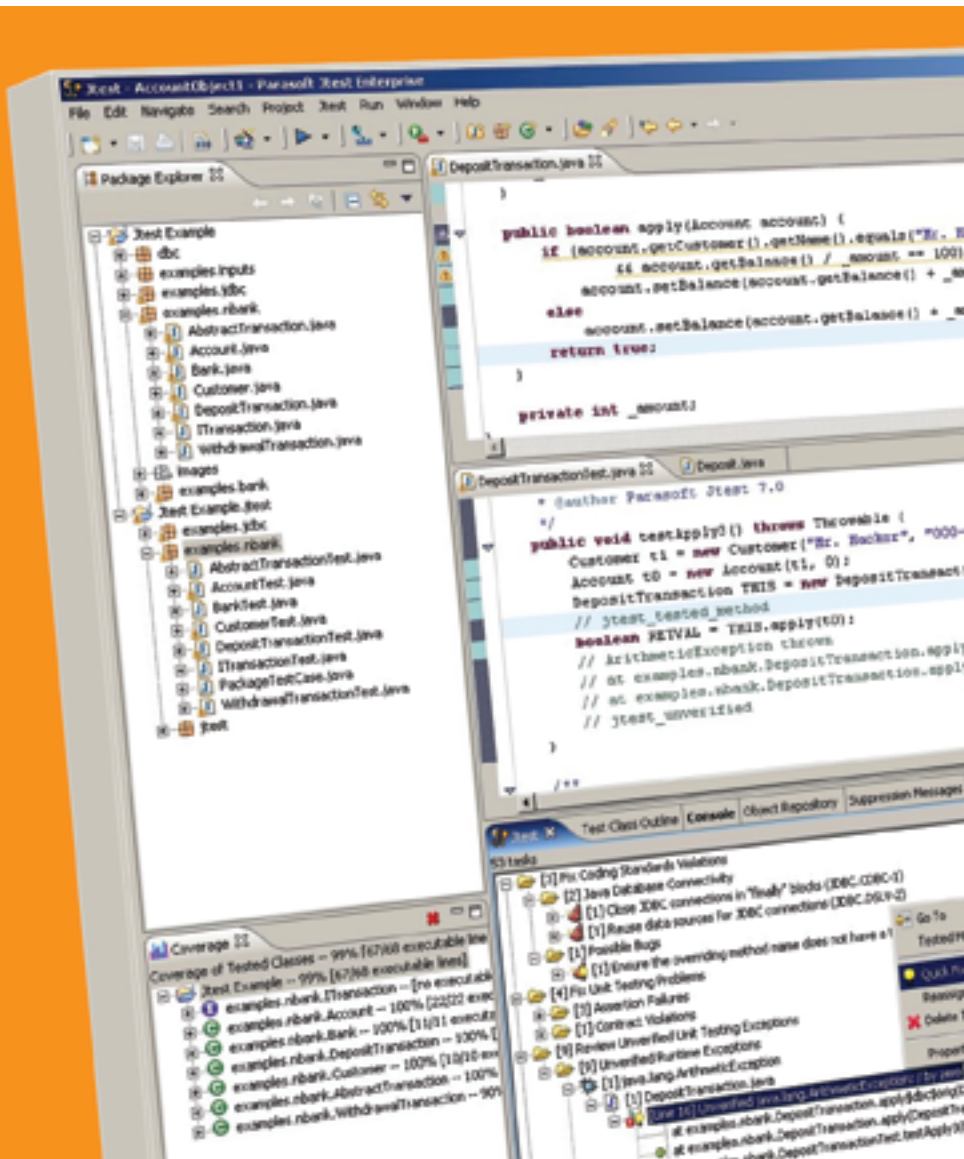


# Code in quality. Test out bugs.

Take control of your code through Automated Unit Testing and Code Analysis...

**PARASOFT®**  
*Jtest*®

- Automatically analyzes your code, applying 500+ Java coding best practices that surface poor code constructs affecting reliability, security and performance.
- Automatically unit tests your Java code evaluating code behavior and exposing unexpected exceptions, boundary condition errors and memory leaks.
- Generates extendable, high coverage test cases in JUnit format that can be quickly turned into libraries of reusable test assets.
- 100's of Quick Fix options for fast, accurate resolution of errors.
- Integrated code coverage analysis ensures that your code is thoroughly tested.
- Team collaboration support allows you to define, distribute and enforce code quality standards across entire development teams.



## Stop bugs before they start:

Jtest automates the valuable, yet often tedious tasks of code reviews and unit testing allowing development teams to adopt a "test as you go" strategy that promotes testing code as it is developed so that quality is built into the code from its inception and bugs eliminated before they infect the rest of the code base.

The result? Cleaner, safer, more reliable and consistent code. Reduced code rework. Faster, more predictable release cycles. Reliable applications and happier, more productive end-users. **To try Jtest today, call 888-305-0041 (x-3501) or go to [www.parasoft.com/JDJmagazine](http://www.parasoft.com/JDJmagazine).**

Parasoft Jtest clients include: Lehman Bros., Cisco, Disney, Raytheon, ADP, Sabre Holdings, GE and more.

**PARASOFT®**  
We make software work.™

**Automated Error Prevention™**