

JAVA DEVELOPER'S JOURNAL

JDJ

MARCH 2004

VOLUME:9 ISSUE:3

Feature:

Java Technology on
Mac OS X



JDJ Industry Profile:

Pursuing the
DREAM

Viewpoint:

IDE Harvesting

Practical Testing:

CLIENT/SERVER
Is Not the Only Fruit

The Oscars of the Software Industry



*JDJ Readers' Choice Awards Recognized
25 Winners and 75 Finalists*

ADAM KOLAWA
OF PARASOFT ON...

THE
FUTURE OF **JAVA**



Adam Kolawa
CEO, Parasoft

RETAILERS PLEASE DISPLAY
UNTIL MAY 31, 2004

\$5.99US \$6.99CAN



04>

0 09281 01751 6

BEA WebLogic Workshop 8.1 makes you more

AMAZING.



Development Tools
December 30, 2003
BEA WebLogic Workshop 8.1
BEA Systems, Inc.

"A shared, consistent development platform...for J2EE experts and business-oriented developers alike."



Java Pro Readers
Choice Award

"WebLogic Workshop empowers all application developers...not just J2EE developers."



Software Development Jolt
Product Excellence Award

"The integration of an IDE, controls and a deployment environment...greatly enhances developer productivity."



CRN Test Center
Product of the Year

"By far the most innovative development tool reviewed this year...when compared with other tools, Workshop blew away the competition."

But don't take their word for it, find out for yourself now, for free.

BEA's WebLogic Workshop 8.1 has won every major award for software development tools in the past year. It's a full-featured Java development environment that lets you visually build and assemble enterprise-scale Web Services, Web Applications, JSPs, Portals, EJBs, and Business Process models based on the latest standards and open source technologies.

Get it now. For free. Visit <http://dev2dev.bea.com/workshop1>

dev2devTM



BUILD JAVA REPORTS WITH NO LIMITS

FORMULA ONE®

Enterprise Reporting Toolsets for Java Projects & Portals

Tightest Integration With J2EE Environments!
No Limits on IDEs or Application Servers!

UNLIMITED productivity
UNLIMITED data access
UNLIMITED report output



Most Flexible License! Extraordinary Value!
No Limits on Savings for Development Teams!

UNLIMITED end users
UNLIMITED projects
UNLIMITED server CPUs



PRODUCTIVE DEVELOPERS • SUCCESSFUL PROJECTS • EMPOWERED USERS

FREE
30
DAY TRIALS

ReportingEngines

JAVA REPORTING TOOLS FROM ACTUATE

WEB: <http://www.ReportingEngines.com>
EMAIL: sales@ReportingEngines.com
TEL: 913-851-2200 • 888-884-8665
FAX: 913-851-1390

FORMULA ONE E.REPORT ENGINE (FOR PDF, HTML, DHTML, AND XML REPORTS)

Unique wizard-driven, point-and-click reporting from Java objects and XML data sources

FORMULA ONE E.REPORT ENGINE FOR WEBLOGIC WORKSHOP

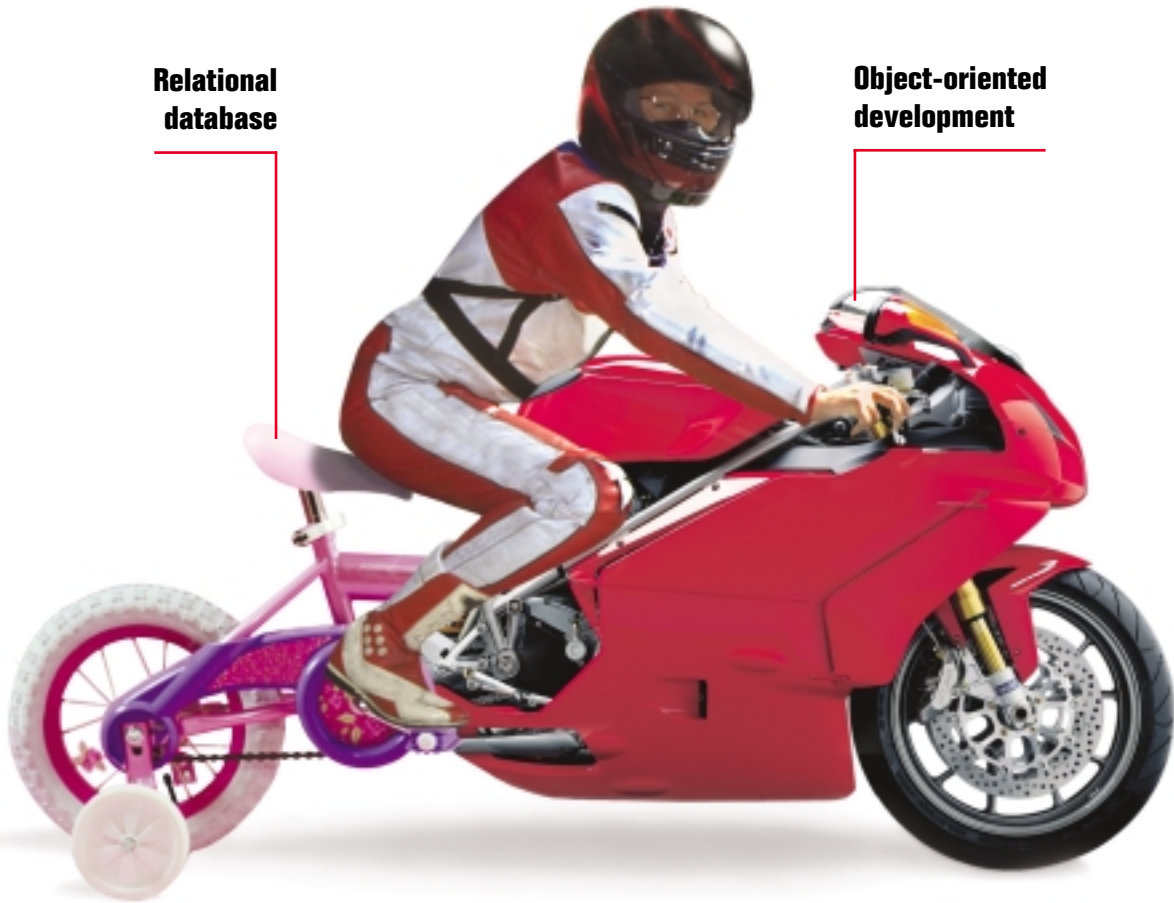
All the power of the regular e.Report Engine as a fully integrated BEA Workshop Control

FORMULA ONE E.SPREADSHEET ENGINE (FOR GENUINE EXCEL REPORTS AND SERVER-SIDE CALCULATIONS)

The financial industry standard for creating & manipulating Excel files in J2EE environments

**Relational
database**

**Object-oriented
development**



A BETTER DATABASE CAN SPEED UP YOUR DEVELOPMENT CYCLE

If your back-end database isn't a good match for your front-end development, you need a new database.

Caché is the *post-relational* database that combines high-performance SQL for faster queries and an advanced object database for rapidly storing and accessing objects. With Caché, no mapping is required between object and relational views of data. That means huge savings in both development and processing time.

Applications built on Caché are massively scalable and lightning-fast. Plus, they require minimal or no database administration.

More than just a database system, Caché incorporates a powerful Web application development environment

that dramatically reduces the time to build and modify applications.

The reliability of Caché is proven every day in “life-or-death” applications at thousands of the world’s largest hospitals. Caché is so reliable, it’s the leading database in healthcare – and it powers enterprise applications in financial services, government and many other sectors.

We are InterSystems, a specialist in data management technology for twenty-five years. We provide 24x7 support to four million users in 88 countries. Caché is available for Windows, OpenVMS, Linux and major UNIX platforms – and it is deployed on systems ranging from two to over 10,000 simultaneous users.



Try a better database. For free.

Download a free, fully-functional, non-expiring version of Caché or request it on CD at www.InterSystems.com/match3

JDJ

contents


Cover Story: **Exclusive**



JDJ Industry Profile... **Adam Kolawa**
 CEO, chairman, and a cofounder of *Parasoft*
 by Jeremy Geelan

54

Special Feature



48
Oracle's Thomas Kurian Pursuing the Dream
 by Jeremy Geelan

FROM THE PUBLISHER

Success, Arrogance, Rise and Fall
 by Fuat Kircaali **6**

FROM THE EDITOR

Can't We All Get Along?
 by Joseph Ottinger **8**

JAVA ENTERPRISE VIEWPOINT

Yet Another IDE War
 by Kirk Pepperdine **10**

AWARDS

'The Oscars of the Software Industry'
JDJ Readers' Choice Awards Recognized 25 Winners and 75 Finalists
 **18**

DRM SOLUTIONS

Provisioning and Digital Rights Management
Instrumentation of J2ME applications
 by Shane Isbell **24**

SECURITY

Chemical Early Warning System in Java
Developing emergency response systems
 by Kathy Lee Simunich, et al **28**

CORE AND INTERNALS VIEWPOINT

Learning to Learn
 by Neil Poska **36**

PRACTICAL TESTING

Client/Server Is Not the Only Fruit
 by Dan North **38**

NEW FEATURES

J2SE 1.5
Tiger debuts with beta
 by Arulazi Dhesiaseelan **44**

DESKTOP JAVA VIEWPOINT

IDE Harvesting
 by Joe Winchester **52**

LABS

IntelliJ IDEA 4.0
by JetBrains, Inc.
 Reviewed by Vijay Phagura **60**

JSR WATCH

From Within the Java Community Process Program
The JVM, J2SE 1.5, and more
 by Onno Kluyt **62**

FROM THE FOUNDING EDITOR

JSF: The Ultimate in Flexibility? Or Complexity?
 by Steve Benfield **64**

@ THE BACKPAGE

Sarbanes-Oxley: The New Rising Star
 by Jack Martin **66**

Features



12
Building a Connected MIDlet Part 2
 by Alex Bourgeois and Richard Kasperowski



40
Clustered Timers
 by Clark Richey, Jr.



58
Java Technology on Mac OS X
 by Daniel H. Steinberg

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.



Fuat Kircaali

Success, Arrogance, Rise and Fall

Last week we were copied in on an internal e-mail from a software company with a market cap in excess of a billion dollars. As I read through this long e-mail thread, which included references to our company, I couldn't help but reflect on the past. I forwarded this e-mail to their CEO and he acknowledged its receipt. History is repeating itself, I realized, and I wanted to share my thoughts with our readers and with the CEO of this software company.

In 1985-86 I worked at IBM as a market research analyst in White Plains, NY, at the company's Information Systems and Communications Group headquarters. Our big boss then was Mike Armstrong. He often held staff meetings in our building at 44 South Broadway. On the day we found out from the *Wall Street Journal* that he had been appointed the head of IBM Europe, he held his last staff meeting there, a farewell meeting. Michael Armstrong was a well-spoken man who inspired his staff. We all speculated that one day he would be the next chief of IBM. He didn't get the top job at IBM, but after a long and successful career he took the helm of another giant ship - AT&T - as the chairman and CEO. After a five-year tenure at AT&T, he is now the chairman of Comcast Corporation.

I learned a lot from Mr. Armstrong and tried to apply this knowledge to my life and career over the next 19 years. Ten years ago when I founded SYS-CON Media, one of the rules I had learned was that arrogance could kill any company in any stage of its life. At the next industry conference, if you don't see me handing out sample copies of our magazines on the show floor, please feel free to accuse me of arrogance.

Over the past 10 years as we've served our *i*-technology markets, we have seen it all. We've worked with giants who were arrogant; we've worked with giants who were focused on their business goals and didn't have time for arrogance. We've worked with very sharp people from organizations both large and small. We've worked with arrogant and stupid people in organizations of all sizes.

What typically happens is that a small startup company comes up with a great new product and gets funding that helps them attain a hockey stick growth curve. In a couple of years they move out of their basement office, go public, and hire thousands of new employees. By all means they deserve their success. What usually goes wrong in the process though is that while the top management remains focused, the company finds itself with a new layer of middle managers. Too young to have an established

corporate culture, the new middle managers with fancy new titles are now in charge of implementing the company's vision. In reality most of them have one goal in mind: When can they cash in their newly acquired stock options?

A few years ago, I remember receiving a phone call from a partner contact in Boston. I drove to his office from New York in three hours rather than simply returning his call, only to have the receptionist tell me that my contact was too busy to see me. I remember stepping into his office and seeing him playing Solitaire. He didn't even recall why he'd left me a message. This incident took place right after this company's IPO and right before their stock options lock-up period ended.

Since we have seen this movie over and over again in the last 10 years, here is how the story line usually develops.

The executives and managers and the company they represent are now too good to do what they had been doing before they found themselves in the midst of this incredible success. Arrogance kicks in. They start treating their partners with arrogance. They start treating their customers with arrogance. They start treating the world with arrogance. Arrogance is the beginning of the end. When the company starts realizing that they are the new miracle golden child of their industry, they also start finding out that they are surrounded by giants. If you look at the history, right after the arrogant phase come two options: either the company gets bought by one of the giants, or they experience a long period of decline and then get bought by a giant at scrap value. This rule doesn't change in hardware, software, anywhere.

I'm not implying that behind the mergers and acquisition news of every software company lies arrogance. I'm saying if you can't really afford to be arrogant, don't use the arrogant giants of your industry as your role model. If you analyze the successes and failures of the giants, you will see that most of their failures come from their arrogance too.

If you lose focus on how you came this far, you'll be faced with the two options I outlined above rather than becoming one of those giants you aspire to be. Don't treat your partners and customers who brought you where you are today with disrespect. Don't let your new middle managers lose your perspective. We would like to see you and your company grow to become one of those giants rather than read rumors everyday about which giant fish will be buying you soon. ☘

International Advisory Board

Calvin Austin	(Sun)
Jason Bell	(Independent)
Jason Briggs	(Independent)
Jeremy Geelan	(SYS-CON)
Thorsten Laux	(Sun)
Rickard Öberg	(Independent)
Joe Ottinger	(Independent)
Bill Roth	(E.piphany)
Ajit Sagar	(Independent)
Eric Stahl	(BEA)
Jon Stevens	(Apache)
Aaron Williams	(JCP)
Alan Williamson	(SYS-CON)
Joe Winchester	(IBM)
Blair Wyman	(IBM)

Editorial

Editor-at-Large:	Alan Williamson
Editor-in-Chief:	Joseph Ottinger
Executive Editor:	Nancy Valentine
Java-Enterprise Editor:	Kirk Pepperdine
Core and Internals Editor:	Jason Bell
Desktop Java Editor:	Joe Winchester
Gaming Editor:	Jason R. Briggs
Contributing Editor:	Ajit Sagar
Contributing Editor:	Glen Cordrey
Founding Editor:	Sean Rhody

Production

Production Consultant:	Jim Morgan
Associate Art Director:	Tami Beatty
Associate Editors:	Jamie Matusow Gail Schultz Jean Cassidy Jennifer Van Winckel
Assistant Editor:	Kelly Flynn
Online Editor:	Lin Goetz
Technical Editor:	Bahadir Karuv, PhD

Writers in This Issue

Jason Bell, Kulvir Singh Bhogal, Todd Brett, Phillip Brittan, Peter Chang, Shigeru Chiba, York Davis, Jeremy Geelan, Kwang Sik Kang, Onno Kluyt, Dan North, Joseph Ottinger, Vijay Phagura, Ingrid Van Den Hoogen, Alan Williamson, Joe Winchester

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

Subscriptions

For subscriptions and requests for bulk orders, please send your letters to Subscription Department 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 © 2004 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 prior permission. For promotional reprints, contact reprint coordinator Carrie Gebert, carrie@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
Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

Newsstand Distribution Consultant
Brian J. Gregory/Gregory Associates/W.R.D.S.
732 607-9941, 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.



WebRenderer™



Standards compliant embeddable web browser component

WebRenderer is a cutting edge embeddable Java™ web content rendering component that provides Java applications and applets with a fast, standards compliant HTML and multimedia rendering engine. WebRenderer provides a feature-packed API including complete browser control, a full array of events, JavaScript interface, DOM access, document history and more.

Why WebRenderer?

- Standards Support (HTML 4.01, CSS 1 & 2, SSL, JavaScript, XSL, XSLT etc.)
- Exceptionally Fast Rendering
- Predictable Rendering
- Scalability (deploy in Applications or Applets)
- Security (based on industry standard components)
- Stability and Robustness

Embed WebRenderer to provide your Java® application with standards compliant web content rendering support.

To download a 30 day trial of WebRenderer visit
www.webrenderer.com

JadeLiquid™
Software

Copyright JadeLiquid Software 2004. JadeLiquid and WebRenderer are trademarks of JadeLiquid Software in the United States and other countries. Sun, Sun Microsystems, the Sun Logo and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks and product names are property of their respective owners.



Joe Ottinger
Editor-in-Chief



Can't We All Get Along?

While browsing through a book on Web services (*XML and Web Services Unleashed* by Ron Schmelzer), some things jumped out at me. First, it's really scary how many options we have in Java. A few months ago Alan Williamson asked, "Haven't We Got Enough to Remember As It Is?" (*JDJ*, Vol. 8, issue 12), and he's dead on – and it gets worse all the time. As the JCP matures and picks up speed, it's easy to be overwhelmed by new paradigms, new configuration choices, new APIs, new acronyms, and new reference implementations – each with their attendant choices of underlying utilities.

For instance, picking up JSR 168 means getting used to Castor, which, of course, is in the same space as Jakarta's commons-digester, which is in the same space as JAXB...and before I forget, it also means learning the portlet API and the mindset that goes with it. It's vastly interesting, and maddening. There are now so many "approved" ways of delivering dynamic content over the Web that it's impossible to make a choice without in-depth analysis of every problem, which lessens the usefulness of the general abstraction.

Second, the names of the suppliers of the APIs stand out. In addition to Sun, of course, you have IBM, Jakarta, then MetaStuff Ltd. (with DOM4J), Oracle, BEA, Caucho...the list goes on forever, it seems. All of these groups have a vested interest in the success of Java as a whole. All of them have code in common use, or have influence on the specifications in (hopefully) beneficial ways. This speaks volumes about a strength of Java – how flexible it is – but also highlights a huge problem.

As many of you are aware, Sun and IBM have a somewhat adversarial relationship with respect to GUI technologies. Sun, as the primary controlling body for Java, advocates their API (Swing) as being what's right for Java. IBM, with an alternate platform

(Eclipse), suggests that SWT is better for clients. There's arguable evidence for both, honestly, but the troubling aspect is the competition.

I'm a very competitive guy. One of the hardest things for me is restraining my expression in appropriate ways when my sons happen to see me lose. Yet I don't want them to think that a loss is a reason for a tantrum. That said, when I'm competing, I don't play stupid games with my opponent; if I win or if I lose, it's because my play had merit or my opponent simply outmaneuvered me, respectively.

The business relationships in Java don't seem to look at things that way. You don't find anyone being less dominant in the industry who simply shrugs and says, "Well, next time we'll win." Instead, we have litigation (e.g., SCO and Linux), acrimony, and outright dislike (e.g., the feeling surrounding the various RSS versions). You have programmers drawing lines in the sand, shouting about ethics and "rightness," as if choosing a platform-native API is a moral decision.

Hey, IBM, Sun, JBoss, BEA, Oracle – it's not. It's a business decision, sometimes swayed by personal preference. It's depressing how many names I could address this to – and how many would still be left out if I committed the space to it.

It's time to drop the acrimony and function as a family whose members compete with each other. We're up against a monopoly that doesn't have the burden and benefit of all of this infighting. If we're up in arms with each other, there's no way we'll ever be able to educate our users enough so they'll really use the power of Java as we know it can be used.

It's time to get along. We can compete forever for mindshare without the internal bickering, and we should – it's good for Java. Tearing each other apart, though, is not. ☹

Joseph Ottinger is a consultant with Fusion Alliance (www.fusionalliance.com) and is a frequent contributor to open source projects in a number of capacities. Joe is also the acting chairman of the *JDJ* Editorial Advisory Board.
josephottinger@sys-con.com



President and CEO:
Fuat Kircaali fuat@sys-con.com
Vice President, Business Development:
Grisha Davida grisha@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
Director, Sales and Marketing:
Megan Ring megan@sys-con.com
Advertising Sales Managers:
Alisa Catalano alisa@sys-con.com
Carrie Gebert carrie@sys-con.com
Associate Sales Managers:
Kristin Kuhnle kristen@sys-con.com
Beth Jones beth@sys-con.com

Editorial

Executive Editor:
Nancy Valentine nancy@sys-con.com
Associate Editors:
Jamie Matusow jamie@sys-con.com
Gail Schultz gail@sys-con.com
Jean Cassidy jean@sys-con.com
Jennifer Van Winckel jennifer@sys-con.com
Online Editor:
Lin Goetz lin@sys-con.com

Production

Production Consultant:
Jim Morgan jim@sys-con.com
Lead Designer:
Tami Beatty tami@sys-con.com
Art Director:
Alex Botero alex@sys-con.com
Associate Art Directors:
Louis F. Cuffari louis@sys-con.com
Richard Silverberg richards@sys-con.com

Web Services

Vice President, Information Systems:
Robert Diamond robert@sys-con.com
Web Designers:
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com

Accounting

Accounts Receivable:
Charlotte Lopez charlotte@sys-con.com
Financial Analyst:
Joan LaRose joan@sys-con.com
Accounts Payable:
Betty White betty@sys-con.com

SYS-CON Events

President, SYS-CON Events:
Grisha Davida grisha@sys-con.com
Conference Manager:
Lin Goetz lin@sys-con.com

Customer Relations

Circulation Service Coordinators:
Shelia Dickerson shelia@sys-con.com
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com
JDJ Store Manager:
Brunilda Staropoli bruni@sys-con.com



Leaner, Meaner, Faster Java Development.

Borland® JBuilder® Developer, from the #1 Java IDE company in the world. It's all the power you crave. Yet lightweight and agile. At a price that won't leave you grounded. Automate the routine stuff. Handcraft the unique. Blast through every stage of the process, with more bullet-proof results. Whether your app is headed to the desktop, Web, or mobile, Borland JBuilder Developer gets you up and going fast. And lands the product flawlessly.

- Customizable code editor with CodeInsight™ and ErrorInsight™
- Import project source from any IDE or editor
- Two-way visual Struts designer
- JSP™ Tag Library/framework support
- Local and remote servlet/JSP debugging
- XML and database tools
- Develop, debug and deploy mobile applications
- Integrated unit testing
- Advanced build and configuration management with Apache™ Ant
- Archive builder
- OpenTools API

Take a test flight today: Sit down, buckle up and hang on at go.borland.com/j3

Made in Borland® Copyright © 2004 Borland Software Corporation. All rights reserved. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. • 21431

Borland®
Excellence Endures™



Kirk Pepperdine
Java Enterprise Editor

Yet Another IDE War

Well, you may have seen the Java Industry Newsletter's hot story, "Eclipse versus NetBeans." As an editor of this fine publication, it's my privilege to see all of our feedback arriving at my inbox. My first reaction was, oh please, not another IDE war. But I dutifully suppressed the urge to delete the e-mails and started to read them.

Most of the posts tried to be helpful by offering advice like, "Eclipse rocks, its user interface feels more natural and is very snappy." There were other comments like "NetBeans rocks, its user interface feels more natural and snappy." Hum...people, I have news for you; it's unlikely that you're going to change my mind on which IDE I should be using just as I am not even going to try to suggest which IDE you should be using. Why, because it's a personal choice, a choice that is influenced by the problems that we are trying to solve and our various idiosyncrasies.

While developing each of these tools, the implementers had to make choices and trade-offs. Each of these decisions (necessarily) both enables and limits what the IDE can do. In addition, it affects the way in which you interact with the tool in order to get it to perform. If someone made the correct choice for you, you most likely wouldn't notice it because you would feel that the

So to all you NetBeans fans, rejoice over Eclipse! It forces NetBeans to improve. And vice versa. Same goes for Swing and SWT. And while all users should be happy about this, let the implementers sharpen their swords and keep on fighting.

Now we have something interesting to talk about, competition. It seems strange to me that Sun, on the one hand, promotes competition in the market place, while on the other hand, they have circled the wagons to protect themselves from competing technologies. The fear that a competing technology may split Java is not without merit. The burning question is: Could the SWT split the Java camp? Going back to the kart analogy, does the fact that Williams builds karts differently than Ferrari split the Formula 1 camp?

Even though Williams and Ferrari do make karts differently, they both must follow a strict set of guidelines or specifications. It is these guidelines that blur the difference in karts to the novice. The reasons for these specifications are somewhat similar in Java as they promote competition. In Formula 1, it's necessary in order to keep the sport interesting to the spectators. The specifications allow for variations,

“ The burning question is: Could the SWT split the Java camp? ”

interaction is natural. If someone made the "wrong" choice, you'd certainly notice because you'd need to think (and consequently question) that particular interaction. In most cases, your choice will be that IDE in which the designers/developers have made the most (or most important set of) "correct" choices.

Beyond this, the impetus is not about which IDE is better. It's a tool, and a tool is just a device to help us construct our end product. As is the case with most professions, the tools need to vary to match the skills and abilities of the professional handling these tools. Sure, a Williams Formula 1 racing kart may look a lot like a Ferrari but, at close inspection, a professional will note all of the things that he likes and dislikes about each kart. Each will be set up differently as they try to solve the same problems but with slightly different techniques and configurations. Even identical karts will be configured slightly differently to take into account the differences in the drivers.

So, it was with great relief when I finally ran across a posting from Thomas Hallgren. His post did not resort to telling my why the IDE of his choice was better than all the others. Instead, he wrote:

We have at least two main contenders in both the open source IDE and the Java GUI market. This is extremely good news. Why? Simply because competition is what drives improvement.

which allow teams to innovate and, hopefully, gain a competitive advantage. About the same can be said for the plethora of Java APIs (such as the J2EE and contained components). What makes the AWT/SWT argument different is that there is no GUI specification in Java. All we have is an implementation. Thus, there is no basis for competition that does not splinter this aspect of Java and hence, the seemingly schizophrenic behavior from Sun. Having said this, it's not so clear that you could apply the same level of abstraction to define a GUI interface as has been done for other components, such as the JMS.

Finally, a post by Jeff Blanton identified the real target, and I quote, "NetBeans is far closer to Microsoft's Visual Basic environment, which is by far one of the best development environments ever made." One thing is for sure, Microsoft is the target and programming in VB does offer some significant advantages over coding in Java. Having said this, even if Microsoft is the target, should we (the Java community) be looking to follow or copy MS? From my perspective, if we do that, then we can never be better than they are. And it is clear, if Java is going to attract the 10 million developers that they are looking for, it will need to be better and IDEs will be an important aspect of that equation. ☺

Kirk Pepperdine is the chief technical officer at [Java Performance Tuning.com](http://JavaPerformanceTuning.com) and has been focused on object technologies and performance tuning for the last 15 years. Kirk is a co-author of *Ant Developer's Handbook* (Sams).

kirk@javaperformancetuning.com



Data connectivity is vitally important to the health of your applications. Is this really where you want to take chances? Your data connectivity choices can have dramatic effects on scalability, interoperability and performance. And you'll be left facing increasing development, maintenance and testing costs plus potential loss of revenue. DataDirect offers the industry's most comprehensive, proven suite of database-independent Type 4 JDBC drivers. Our extensively tested J2EE-certified drivers include the most advanced JDBC 3.0 features including Distributed Transaction Support, Connection Pooling and BLOB/CLOB support. DataDirect Connect[®] for JDBC[®] is the SPECjAppServer and ECperf performance leader.

DataDirect[™]
TECHNOLOGIES

www.datadirect.com
800-876-3101

Find out what else you might be missing. Download our whitepaper, "What you don't know about database drivers CAN hurt you" @ www.datadirect.com/JDJ

BUILDING A CONNECTED MIDlet

...and writing
a full-featured
J2ME application

Part 2

by Alex Bourgeois and Richard Kasperowski

This article is the second in a two-part series devoted to developing a connected MIDlet. In the first part of the series, we designed and built a functional application that collected a string from the user, established a connection to a remote server, sent and received data along that connection, and then presented the result to the user. In this article, we'll refine the design of the application, add a few features that are essential to any connected MIDlet, and implement exception handling.

This article assumes you have read "Building a Connected MIDlet, Part 1" (*JDJ*, Vol. 8, issue 12) and installed our Echo application. In case you need them, the servlet .ear file and the rest of the code are still available for download at www.altisimo.com/download/echo.ear and www.altisimo.com/download/echo-1.0.zip.

By the end of this article, you'll have learned how to:

- Break a single class application into better partitioned packages and classes
- Invoke a new method in a background thread
- Interrupt an ongoing background thread
- Gracefully handle application-specific exceptions

This is the list of issues from the first article that we still need to address:

- **Architecture:** Our client is complex enough that we should be using multiple classes to handle the various tasks. We will rework the code to follow the architecture of Sun's J2ME blueprint.
- **Progress indicator:** Whenever users select a command that takes more than a second to complete, we need to let them know that the application is working on it. In Echo, we should tell the user how far along we are in the network transaction between the mobile device and the servlet. In the current implementation, the user sees a blank screen during the network round-trip and has no idea what is happening.
- **Stop command:** We should allow the user to interrupt any task that takes more than a second to complete. In the case of Echo, this again is the network transaction. Users currently have no way to stop the network round-trip to check their calendar or do anything else on the handheld device.
- **Input validation:** The J2EE component of our application will throw an exception if the string it receives is null or empty. One way to address this issue is to add a preliminary client-side test before we send the data to the servlet.
- **Exception handling:** There are a variety of other issues that can arise during the execution of our application. We'll explore how to identify and handle many of these issues.

Architecture

We based the second iteration of Echo on Sun's J2ME Smart Ticket blueprint, using the same architecture and class names whenever possible, while removing any unused features. This makes it easier to become familiar with the inner workings of the blueprint in a simplified context. The server-side portion of this application has not changed from the first version. It's a simple servlet that reads in a string from the client and returns the same string. The server side consists of the following classes:

- **EchoServlet:** The servlet class, which reads in the string using `readUTF` and then writes it back out using `writeUTF`.
- **EchoTest:** An automated test client to ensure that the servlet is functioning properly before we begin writing J2ME client code. For a more complex application, we would use JUnit, but for this example this simpler test is acceptable.

The J2ME client-side portion of this version of Echo looks more complex on the surface. We start by thinking about the application from the Model-View-Controller (MVC) point of view. Figure 1 illustrates the basic MVC architecture and how the different layers interact.

We'll use three packages, each of which encapsulates part of the MVC architecture; our complete architecture is shown in Figure 2. It might seem extreme to make the Echo client so complex, but we can reuse this architecture and these packages in future projects. The packages are:

- **echo.model:** Contains the model part of the MVC architecture – the classes that model what the client needs to know about the server side of the application. This includes data objects, business logic, and a proxy connector to the server.
- **echo.ui:** Contains the view and the controller parts of the MVC architecture. These are the classes that obtain and display information from and to the user, and control the flow from screen to screen.
- **echo.util:** Contains the utility classes that are used by both the model and UI packages.
- **echo:** A top-level package containing exactly one class, `EchoMIDlet`. This is the entry point for the application. `EchoMIDlet` creates instances of the model, the view, and the controller, and delegates control to the controller.

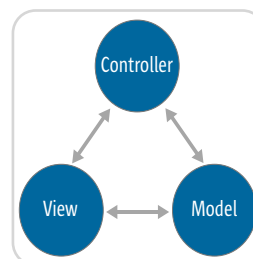


Figure 1 MVC architecture diagram

The model package holds all the business objects used on the mobile side of the application, allowing us to separate business logic from the user interface classes. The model package contains two classes:

- **RemoteModelProxy:** An instance of the Proxy design pattern, it offers a local method call interface to the remote server. RemoteModelProxy is responsible for establishing a connection with the remote server, passing on any requests made by the user, and receiving the data returned by the remote server. This is where the actual communication happens, so this is also a place where we must catch the many exceptions that might be thrown, and present the user with a message that explains what has happened – why the application was unable to complete the requested task and what the user should do differently to avoid the error in the future.
- **ModelFacade:** We use the Facade pattern to control access to the model package. ModelFacade is the single access point for other classes into the model of the program. It creates a RemoteModelProxy as an instance variable, and then delegates all subsequent requests onto it.

The UI package contains the view and controller parts of the MVC architecture: all of the screens used to gather and display information, as well as the classes required to manage them. In Echo, the view consists of three forms, and the controller is implemented by a UIController object.

- **GetInputUI** is a form that collects a single string from the user. Figure 3 shows GetInputUI immediately after the MIDlet is started.
- **ProgressObserverUI** holds a gauge that displays the progress made in the network transaction. This class implements the ProgressObserver interface. Figure 4 shows the ProgressObserverUI in the middle of the connection process.
- **ShowOutputUI** is another form. It displays the server's response to the user, as in Figure 5.
- **UIController** handles the transitions between screens. It performs the actions that ensure that the user is presented with correct and up-to-date information as to what is happening in the application. This class also spawns and starts a new thread to display the ProgressObserverUI.

The util package is used by classes in both the model and the UI packages, thereby maintaining separation between the other packages while not requiring repeated code to handle exceptions and other assistive tasks, such as displaying a progress indicator.

- **ApplicationException** encapsulates any exceptions that might be thrown during the execution of the application.
- **ErrorMessageCode** is an additional level of abstraction used to categorize some of the ApplicationException.
- **ProgressObserver** is an interface implemented by ProgressObserverUI. It defines the basic methods required to display and control the gauge. ProgressObserver contains methods to update the gauge, set the “is stoppable” flag, determine if the stoppable flag is set, and stop the ongoing process.

Now that you're familiar with how the classes are packaged and their basic functionality, let's look at how we implemented some of the improvements to the application.

Progress Indicator

A powerful aspect of the J2ME platform is its inherent multi-threading capability. By using multiple threads, an application can perform a time-consuming task in the background while still updating the UI to keep the user informed. There's nothing more frustrating than hitting a button and not having any idea

whether anything is going on: Is the application doing something difficult or did it just hang?

The only part of Echo that's likely to take some time to complete a task is in the method RemoteModelProxy.echo(). This is where the client creates and communicates along the connection to the remote server. While the device is establishing a connection and sending data, there is currently no visual feedback to let the user know that anything is happening. This is why we will implement a Progress Indicator: to let the user know where the application is in the connection process. The ProgressObserver runs in its own thread and uses the ProgressObserverUI to provide information to the user. ProgressObserverUI extends the form and uses a gauge to show how far along in its task the ProgressObserver is.

In our implementation, when the user wants to send out the string he or she entered, we call UIController.doEcho(). doEcho() spawns the new thread and then passes the thread and the user's string as arguments to runWithProgress(). runWithProgress() prepares the ProgressObserverUI, sets it as the current screen, and starts the new thread. The thread executes the code in run() (see Listing 1).

model.echo() actually calls RemoteModelProxy.echo(), which establishes the connection to the remote server, transmits the string the user provided, and returns the response from the server (see Listing 2).

By creating a separate thread for the ProgressObserver and setting it to be the current display before we start to connect to the remote server, we let RemoteModelProxy do the complex work of establishing and using the connection in the background. This means we can update the ProgressObserverUI to let the user know the state of the application. Once RemoteModelProxy.echo() has returned, the result is appended to ShowOutputUI, and ShowOutputUI is set to be the current display.

Model-View-Controller Architecture

The Model-View-Controller (MVC) Architecture divides the functionality of an application into three layers: model, view, and controller. MVC decouples the responsibilities of the different layers and assigns each layer a specific set of tasks.

A model represents business objects and the rules as to how they are accessed and updated. The model informs views when data changes and contains methods for the views to determine its current state.

A view displays the contents of a model to the user. It's responsible for how the data is presented. Views also forward user actions to the controller.

A controller directs the actions within the application. It's responsible for interpreting user input and triggering the appropriate model actions.

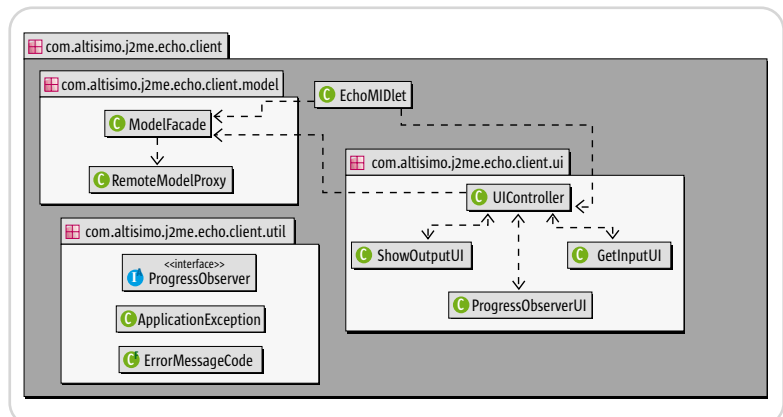


Figure 2 Echo class diagram

Notice that after each important step in `echo()` we make a call to `checkAndUpdateProgress()`. This allows us to break the method into relevant chunks, and perform any tests required before continuing through the method.

```
protected void checkAndUpdateProgress() throws
ApplicationException {
    if (progressObserver != null) {
        if (progressObserver.isStopped()) {
            throw new ApplicationException(
                ErrorMessageCode.ERROR_OPERATION_INTERRUPTED);
        }
        else {
            progressObserver.updateProgress();
        }
    }
}
```

`checkAndUpdateProgress()` makes sure there is a `ProgressObserver` to update, and tests to see if the `ProgressObserver`'s `stopped` flag has been set to true. If the flag is not set, we increment the gauge and continue on through the method. If the flag is set, we throw an `ApplicationException` and exit the method. Setting the `stopped` flag is handled by the `stop` command.

Stop Command

The `ProgressObserver` is a nice feature, but what if the user needs to cancel a transaction while communicating with the remote server? We need to give him or her a way to cancel a task before it returns. This is a situation for a `stop` command.

We have already added commands to our application in `UIController`, but in this case we are using a command that is local to the `ProgressObserverUI`. All of the previously implemented commands have been used when the program was inactive and waiting for user input to initiate a new task. In this case, we want to stop an ongoing task and to have that task interrupted as quickly as possible.

Look back at the implementation of `echo()` and notice that we delegate all of the calls to increment the gauge to `checkAndUpdateProgress()` when `RemoteModelProxy` completes a step in the communication process. We do not have a way to instantly shut down the `RemoteModelProxy` as it is contacting the remote server, but we can have the `RemoteModelProxy` check if it should continue before each additional step. This is where we use the `stop` command, and it is the reason `ProgressObserverUI` implements `CommandListener`.

When the `stop` button is pressed, the `stopped` flag in the `ProgressObserver` is set to true, and the `RemoteModelProxy` will stop executing `echo()` the next time `checkAndUpdateProgress()` is called. `checkAndUpdateProgress()` will throw an

`ApplicationException` of type `ErrorMessageCode.ERROR_OPERATION_INTERRUPTED` to let the caller know what happened:

```
public void commandAction(Command command, Displayable display)
{
    if (command == stopCommand) {
        stopped = true;
    }
}
```

We can then catch the `ApplicationException` in `UIController.run()` and display a message to the user letting him or her know that they successfully interrupted the communication process.

Input Validation

If we send a null or empty string to the server, our application will throw a `RuntimeException` when `EchoServlet` calls `readUTF()`. We need to implement some very basic input validation in the client before we call `doEcho()`. It would be an easy enough task to change `EchoServlet` to handle this case, but we don't want to take the time and resources to create a connection if there is no useful information to send along it. Remember, our handheld device is limited – its network connection is likely to be slow and have high latency – and many carriers charge by the packet. This is a case where a few lines of additional code in the client can save the user some time and money. Because the buttons in `GetInputUI` use `UIController` as their `CommandListener`, we have `commandAction()` call `GetInputUI.isValidInput()` to ensure that the user has provided an acceptable string. By having the callback to `GetInputUI` for the validation, we can customize `isValidInput()` for future versions of `GetInputUI` without having to change `UIController` (see Listing 3).

If the user has not provided valid input, `GetInputUI.tryAgain()` is called. The user is then prompted to enter a string again, after being shown an error telling him or her that the input cannot be blank.

```
void tryAgain() {
    uiController.showErrorAlert(
        new
ApplicationException(ErrorMessageCode.ERROR_BLANK_INPUT),
        this);
}
```

This is about as thorough as client-side data validation needs to be. A nice side effect is that it teaches the user how to use the application.

Exception Handling

Our exception handling scheme is also based on the architecture from the Sun blueprint. We use `ApplicationException` to handle all nonnative exceptions, and use constants within `ErrorMessageCode` to store the various types of `ApplicationExceptions` that can occur most frequently (see Listing 4).

`ApplicationException` has a no-argument and three single-argument constructors that take either a string, an int, or another exception. These allow you to set your own error message, use one of the predefined messages, or wrap another exception in an `ApplicationException` to pass it up the chain of method calls.

When the int-argument constructor is used, the int is mapped to one of the existing error types in `ErrorMessageCode`. The corresponding text is assigned to the message attribute of the `ApplicationException` instance, with a default error message to handle any undefined errors that are encountered (see Listing 5).

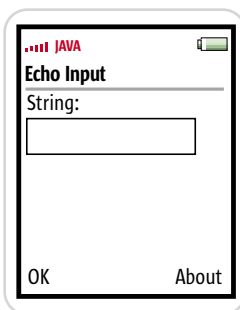


Figure 3 GetInputUI

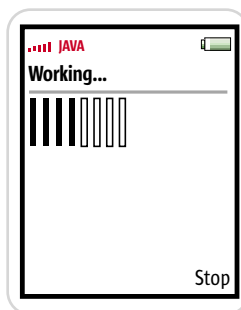


Figure 4 ProgressObserverUI

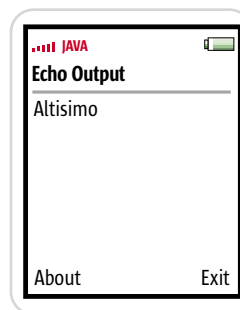


Figure 5 ShowOutputUI



Optimize J2EE.

The J2EE revolution is here to increase performance, value, and lower IT costs. It's a solution from Mercury Interactive that makes your whole J2EE applications ecosystem work right.

It's the fastest way to find the toughest J2EE problems from the development to the live applications. Even pinpoints root cause and the line of source code.

It's about more than delivering J2EE applications. It's about delivering applications that work and yield real business value.

It's the Business Technology Optimization revolution.

And Mercury is the only one bringing you a revolutionary solution for J2EE.

Download our free white paper, "**Diagnosing J2EE Performance Problems Throughout the Application Lifecycle,**"

at www.mercuryinteractive.com/optimizej2ee

Get Optimized™



Each `ApplicationException` has a string attribute and an exception attribute, and when either of the other single-argument constructors are used, the corresponding instance attribute is set to the argument that is passed into the constructor.

```
public ApplicationException(String msg) {
    this.message = msg;
    return;
}

public ApplicationException(Exception exception) {
    this.exception = exception;
    return;
}
```

By having a single, versatile `ApplicationException`, we can have greatly simplified catch blocks, but still be sure that we are indeed handling all possible nonnative exceptions by

catching only `ApplicationException`, and that we are still able to customize error messages for any situation we might encounter.

Conclusion

We have now written a complete example of how to connect to and communicate with a remote server. There are more features we could add to this application to offer the user more control or to handle more obscure potential exceptions, but in its current state, Echo is more than enough for a sample connected application. We use Echo as the basis for more complex applications, and we hope that this article provides you with a foundation to develop your own connected MIDlets.

There are many more issues to consider when writing a full-featured J2ME application, such as accessing the device's memory store or being able to operate in a disconnected mode, and we look forward to covering them in upcoming articles. If there are any topics that you want to see included in the series, please send us an e-mail at info@altisimo.com.

Listing 1

```
private void doEcho() {
    Thread thread = new Thread(this);
    userInput = getInputUI.getInputText();
    runWithProgress(thread, "Working...", true);
}

private void runWithProgress(Thread thread,
    String title,
    boolean stoppable) {
    progressObserverUI.init(title, stoppable);
    setCurrent(progressObserverUI);
    thread.start();
}

public void run() {
    try {
        String result = model.echo(userInput);
        showOutputUI.setResult(result);
        showOutputUI.addFields();
        setCurrent(showOutputUI);
    } catch (ApplicationException ae) {
        switch(ae.getCode()) {
            case ErrorMessageCode.ERROR_OPERATION_INTERRUPTED:
                showOutputUI.setResult("[operation terminated by user]");
                showOutputUI.addFields();
                showInfoAlert(null, ae.getMessage(), showOutputUI);
                break;
            ...
        }
    }
}
```

Listing 3

```
public void commandAction(Command command, Displayable displayable) {
    if (command == okCommand && displayable == getInputUI) {
        if (getInputUI.isValidInput()) {
            doEcho();
        } else {
            getInputUI.tryAgain();
        }
        ...
    }

    boolean isValidInput() {
        String input = getInputText();
        return !(input == null || input.equals(""));
    }
}
```

Listing 4

```
/** Use this error code if a more specific one does not exist. */
public static final int ERROR_GENERAL = 0;

/** Indicates a communication failure between the client and remote server. */
public static final int ERROR_COMM_FAILURE = 1;

/** The user did not enter any text in the input screen. */
public static final int ERROR_BLANK_INPUT = 2;

...
}
```

Listing 5

```
public ApplicationException(int code) {
    this.code = code;

    switch(code) {
        case ErrorMessageCode.ERROR_GENERAL:
            setMessage("An ApplicationException has occurred.");
            break;

        case ErrorMessageCode.ERROR_COMM_FAILURE:
            setMessage("A communication error has occurred.");
            break;

        case ErrorMessageCode.ERROR_OPERATION_INTERRUPTED:
            setMessage("The current operation has been interrupted.");
            break;

        case ErrorMessageCode.ERROR_BLANK_INPUT:
            setMessage("User input cannot be blank.");
            break;

        default:
            setMessage("An unidentified error has occurred.");
            break;
    }
}
```

Listing 2

```
public String echo(String input) throws ApplicationException {
    try {
        HttpURLConnection con = openConnection();
        checkAndUpdateProgress();

        DataOutputStream dataOutputStream = openConnectionOutputStream(con);
        dataOutputStream.writeUTF(input);
        checkAndUpdateProgress();

        // Get the response from the server.
        DataInputStream dataInputStream = openConnectionInputStream(con);
        String fromServer = null;
        fromServer = dataInputStream.readUTF();
        checkAndUpdateProgress();

        closeConnection(con, dataOutputStream, dataInputStream);
        return fromServer;
    } catch (IOException ioe) {
        throw new ApplicationException(ErrorMessageCode.ERROR_COMM_FAILURE);
    }
}
```



Alex Bourgeois is a cofounder of Altisimo Computing, a J2ME consulting and training company. He has been programming in Java since 1997 and has developed numerous applications for private and commercial use. Alex also cochairs the New England Java Users Group J2ME Special Interest Group.



Richard Kasperowski, a cofounder of Altisimo Computing, has been programming in Java since 1997 and has developed numerous applications for private and commercial use. Richard is also the cochair of the New England Java Users Group J2ME Special Interest Group.

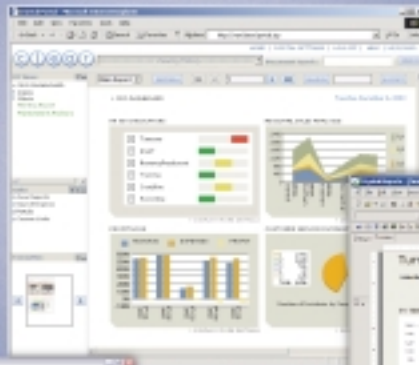
jdj@altisimo.com

99.9% of the world won't find these
screen shots terribly exciting.
But if you're in the other 0.1%,
yeehaw.

Use Crystal Reports 10 with
your J2EE applications }

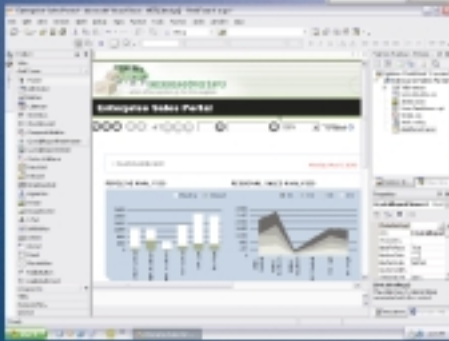
New 100% Java reporting component. Deploy
reports across Unix, Linux and Windows

Extend Crystal Reports with Crystal Enterprise.
Get world-class web report publishing,
management, and scalability



Visual Designer simplifies
data connectivity
Deliver diverse data formatting
options within your presentation layer

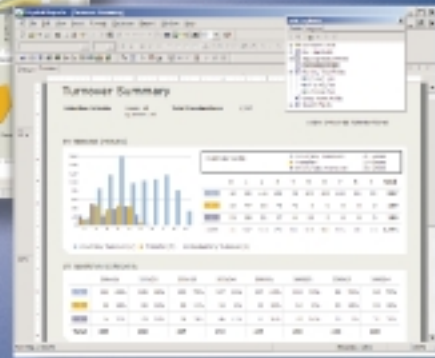
Access data natively, or via
ODBC, JDBC and OLE DB



{ Expedite .NET and
J2EE report integration

Design and integrate reports
from within popular IDEs

Flexible Java, .NET and COM SDKs support the
tight integration of report interactivity including:
group tree navigation, exporting, printing, and
drill down



New Crystal Reports 10.

The best in business intelligence now offers the best in business reporting.
New Crystal Reports® 10 is a faster and simpler way for developers to integrate dynamic
data into applications and implement high-quality viewing, printing, and exporting. Learn
more about Crystal Reports 10 and Crystal Enterprise™ 10, and access technical and evaluation
resources at www.businessobjects.com/v10/047 or contact us directly at 1-800-877-2340.



'The Oscars of the Software Industry'

JDJ Readers' Choice Awards Recognized 25 Winners and 75 Finalists



Widely referred to as the "Oscars of the software industry," the

JDJ Readers' Choice Awards program has become the most-respected

industry competition of its kind.

The polls were open for just under a year, from March 1, 2003, through December 31, 2003, and over 15,000 JDJ readers cast their votes.

JDJ Readers' Choice Award recipients are selected through reader-submitted nominations, followed by online voting.



BEST JAVA BOOK

Winner: *Thinking in Java*

by Bruce Eckel

Prentice Hall PTR www.phptr.com

Thinking in Java, Third Edition, teaches Java one step at a time, with hundreds of new, self-contained example programs that illuminate every key technique for building today's applications. This edition has been fully updated for Java 1.4, with a new extended coverage of multithreading, and many other major improvements.

1st Runner-Up: *Java Development with Ant*

by Erik Hatcher and Steve Loughran

Manning Publications Co. www.manning.com

2nd Runner-Up: *Core J2EE Patterns: Best Practices and Design Strategies*

by Deepak Alur, Dan Malks, and John Crupi

Prentice Hall PTR www.phptr.com

3rd Runner-Up: *About Face 2.0: The Essentials of Interaction Design*

by Alan Cooper and Robert M. Reimann

Wiley Publishing www.wiley.com



BEST DATABASE TOOL OR DRIVER

Winner: *AllFusion ERwin Data Modeler*

Computer Associates International, Inc. <http://ca.com>

AllFusion helps users increase productivity in the design, generation, and maintenance of high-quality, high-performance databases, data warehouses, and enterprise data resource models.

1st Runner-Up: *XMLSPY*

Altova www.altova.com

2nd Runner-Up: *IBM Rational Rose Data Modeler*

IBM www.ibm.com

3rd Runner-Up: *Oracle TopLink*

Oracle Corporation www.oracle.com



BEST ENTERPRISE DATABASE

Winner: *Oracle9i Database*

Oracle Corporation www.oracle.com

Oracle9i Database is a scalable and full-featured database. It features full XML database functionality with the Oracle XML DB feature, enhancements to Oracle9i Real Application Clusters, and additional self-tuning and self-management capabilities to help improve DBA productivity and efficiency.

1st Runner-Up: *DB2 Universal Database v 8.1*

IBM www.ibm.com

2nd Runner-Up: *JDataStore 6*

Borland Software www.borland.com

3rd Runner-Up: *Advantage Ingres Enterprise Relational Database*

Computer Associates <http://ca.com>



BEST JAVA APPLICATION

Winner: *Borland Together ControlCenter*

Borland Software www.borland.com

Borland Together ControlCenter is a complete application development environment that encompasses and integrates application design, development, and deployment, simplifying the creation of complex software applications through its model-build-deploy platform. Through its integrated and intuitive development environment, ControlCenter brings cross-functional teams together using common language, diagrams, and building-block components.

1st Runner-Up: *IBM WebSphere Application Server v5.0*

IBM www.ibm.com

2nd Runner-Up: *IntelliJ IDEA*

JetBrains Inc. www.intellij.com

3rd Runner-Up: *BEA WebLogic Server 7.0*

BEA Systems www.bea.com



BEST JAVA APPLICATION SERVER

Winner: *BEA WebLogic Server*

BEA Systems www.bea.com

BEA WebLogic Server 7.0 simplifies deployment of massive clustered applications by providing configuration wizards and two-phase deployment capabilities. Configuration wizards are available to create new domains, clusters, and configurations. The configuration

wizard reduces the complexity involved in the deployment process by navigating the user through multiple configuration features and validating each step in the configuration process.

1st Runner-Up: *JBoss 3.x Application Server*

JBoss Group, LLC www.jboss.com

2nd Runner-Up: *IBM WebSphere Application Server v5.0*

IBM www.ibm.com

3rd Runner-Up: *Fiorano Enterprise Service Bus (ESB)*

Fiorano Software, Inc. www.fiorano.com



BEST JAVA COMPONENT

Winner: *Eclipse SWT*

Eclipse.org www.eclipse.org

Eclipse is an open platform for tool integration built by an open community of tool providers. Operating under an open source paradigm, with a common public license that provides royalty-free source code and world-wide redistribution rights, the Eclipse platform provides tool developers with flexibility and control over their software technology. Eclipse was invented by IBM but is now an open-source development platform governed by 30 software vendors and the open source community.

1st Runner-Up: *WebObjects 5.2*

Apple www.webobjects.com

2nd Runner-Up: *Oracle9i JDeveloper - BC4j*

Oracle Corporation www.oracle.com

3rd Runner-Up: *BouncyCastle Crypto APIs*

Legion of the Bouncy Castle www.bouncycastle.org



BEST JAVA DATA ACCESS TOOL

Winner: *IBM WebSphere Studio (Application Developer v5.0)*

IBM www.ibm.com

IBM WebSphere Studio is an open, comprehensive development environment for building, testing, and deploying dynamic on-demand e-business applications. Founded on open technologies and built on Eclipse, WebSphere Studio provides a flexible, portal-like integration of multilanguage, multiplatform, and multidevice application development tools that maximize developer productivity, increase ROI, and improve overall time to value.

1st Runner-Up: *Fiorano ESB Data Access Service*

Fiorano Software, Inc. www.fiorano.com

2nd Runner-Up: *Kodo JDO*

SolarMetric www.solarmetric.com

3rd Runner-Up: *Oracle TopLink*

Oracle Corporation www.oracle.com



BEST JAVA DEBUGGING TOOL

Winner: *NetBeans IDE*

Sun Microsystems, Inc. www.sun.com

The NetBeans IDE is a development environment written in Java that can be used to develop code in Java, HTML, XML, JSP, C/C++, and other languages. The IDE is modular, and there are a variety of commercial and free extensions to it to support various technologies.

1st Runner-Up: *IBM WebSphere Studio (Application Developer v5.0)*

IBM www.ibm.com

2nd Runner-Up: *Quest JProbe*

Quest Software, Inc. www.quest.com

3rd Runner-Up: *Oracle9i JDeveloper*

Oracle Corporation www.oracle.com



BEST JAVA IDE ENVIRONMENT

Winner: *JBuilder*

Borland Software Corporation www.borland.com

Borland JBuilder is a cross-platform environment for building enterprise Java applications. JBuilder 8 Enterprise simplifies Web and EJB development with two-way visual designers and rapid deployment to leading J2EE platform application servers. Enhance productivity with UML code visualization, refactoring, code formatting, HotSwap debugging, unit testing, and version control integration.

1st Runner-Up: *Eclipse*

Eclipse.org www.eclipse.org

2nd Runner-Up: *IBM WebSphere Studio v5.0*

IBM www.ibm.com

3rd Runner-Up: *IntelliJ IDEA*

JetBrains Inc. www.intellij.com



BEST JAVA INSTALLATION TOOL

Winner: *InstallAnywhere Enterprise Edition v5*

Zero G Software, Inc. www.zerog.com

InstallAnywhere was designed to install any kind of software – desktop, enterprise, or multitiered Web services – onto any platform, and configure those applications for optimal performance. Whether you're creating applications written in Java, C++, Perl, or C#, or deploying EJBs or .NET Web services, InstallAnywhere automatically handles all of the installation details.

1st Runner-Up: *InstallShield MultiPlatform*

InstallShield Software Corporation www.installshield.com

2nd Runner-Up: *Quest DeployDirector*

Quest Software, Inc. www.quest.com

3rd Runner-Up: *Venus Application Publisher (Java Web Start Ed.)*

Venus Labs www.vamphq.com



BEST JAVA MESSAGING TOOL

Winner: *FioranoMQ*

Fiorano Software, Inc. www.fiorano.com

FioranoMQ is a scalable, secure, and fast Java Messaging Server for organizations building scalable, high-performance, and reliable distributed computing solutions. Fiorano's configurable, multilevel security system adds Java REALMS support to improve security management by centralizing and combining user authentication sources. The latest release of FioranoMQ is compliant with the JMS 1.1 Specifications released by Sun Microsystems.

1st Runner-Up: *IBM WebSphere MQ 5.3*

IBM www.ibm.com

2nd Runner-Up: *SwiftMQ 4.0*

IIT Software www.swiftmq.com

3rd Runner-Up: *Oracle9i Application Server*

Oracle Corporation www.oracle.com



BEST JAVA PERSISTENCE ARCHITECTURE

Winner: Oracle TopLink

Oracle Corporation www.oracle.com

Oracle TopLink is an object-to-relational persistence framework that is suitable for a wide range of J2EE and Java application architectures. TopLink enables the creation of high-performance applications that store persistent data in relational databases. TopLink simplifies the access to relational data without sacrificing performance or functionality.

1st Runner-Up: Kodo JDO

SolarMetric www.solarmetric.com

2nd Runner-Up: Describe

Embarcadero Technologies www.embarcadero.com

3rd Runner-Up: The Open For Business Project

The Open For Business Project www.ofbiz.org



BEST JAVA PROFILING/TESTING TOOL

Winner: JProfiler

ej-technologies www.ej-technologies.com

JProfiler is a dynamic profiler based on the Java Virtual Machine profiling interface (JVMPi). It's targeted at J2EE and J2SE applications, and features CPU profiling, memory profiling, thread profiling, and VM telemetry information.

1st Runner-Up: Quest JProbe

Quest Software, Inc. www.quest.com

2nd Runner-Up: IBM WebSphere Studio (Application Developer v5.0)

IBM www.ibm.com

3rd Runner-Up: Oracle JDeveloper

Oracle Corporation www.oracle.com



BEST JAVA TRAINING PROGRAM

Winner: IBM developerWorks

IBM www.ibm.com

IBM developerWorks helps users unleash the power of hardware and operating systems, bridge multiple platforms, and be more successful through the use of open standards and cross-platform technologies such as Java, Linux, XML, and open source projects.

1st Runner-Up: BEA dev2dev

BEA Systems www.bea.com

2nd Runner-Up: Oracle University - Java Developer Learning Paths

Oracle Corporation www.oracle.com

3rd Runner-Up: Java BluePrints Program

Sun Microsystems www.sun.com



BEST JAVA VIRTUAL MACHINE

Winner: BEA WebLogic JRockit

BEA Systems www.bea.com

BEA WebLogic JRockit is a high-performance Java Virtual Machine developed for server-side applications and optimized for Intel architectures designed to ensure reliability, scalability, manageability, and flexibility for Java applications. A component of the BEA WebLogic Enterprise Platform, WebLogic JRockit delivers a new level of performance for Java applications deployed on any kind of hardware architecture.

1st Runner-Up: IBM Developer Kit, Java 2 Technology Edition, v 1.4.0

IBM www.ibm.com

2nd Runner-Up: Oracle JVM

Oracle Corporation www.oracle.com

3rd Runner-Up: Excelsior JET

Excelsior, LLC www.excelsior-usa.com



BEST JAVA WEB SERVICES TOOLKIT

Winner: BEA WebLogic Workshop

BEA Systems www.bea.com

BEA WebLogic Workshop is an integrated development framework that enables application developers to create, test, and deploy enterprise-class Web service applications on the BEA WebLogic Platform. It provides a unified development platform that enables developers to build and connect components, data, and application business logic, while insulating them from the complexities of J2EE.

1st Runner-Up: IBM WebSphere SDK for Web Services

IBM www.ibm.com

2nd Runner-Up: Oracle JDeveloper

Oracle Corporation www.oracle.com

3rd Runner-Up: Advantage Plex

Computer Associates <http://ca.com>



BEST MOBILE DATABASE

Winner: PointBase Micro 4.5

PointBase, Inc. www.pointbase.com

A high-performance Java RDBMS that has an ultrasmall footprint of 46K, perfect for advanced Java-based mobile devices. Designed for the J2SE and J2ME platforms, PointBase Micro 4.5 supports a broad subset of the JDBC API. With its all-Java implementation, PointBase Micro 4.5 enables a wide variety of enterprise applications to run on today's growing number of small mobile devices, eliminating the need for a constant wireless connection to enterprise database servers.

1st Runner-Up: Oracle JI Lite

Oracle Corporation www.oracle.com

2nd Runner-Up: SQL Anywhere Studio

iAnywhere Solutions <http://ianywhere.com>



BEST TEAM DEVELOPMENT TOOL

Winner: BEA WebLogic Workshop

BEA Systems www.bea.com

BEA WebLogic Workshop is an integrated development framework that enables application developers to create, test, and deploy enterprise-class Web service applications on the BEA WebLogic Platform. It provides a unified development platform that enables developers to build and connect components, data, and application business logic, while insulating them from the complexities of J2EE.

1st Runner-Up: IBM WebSphere Studio v5.0 with Rational Clearcase

IBM www.ibm.com

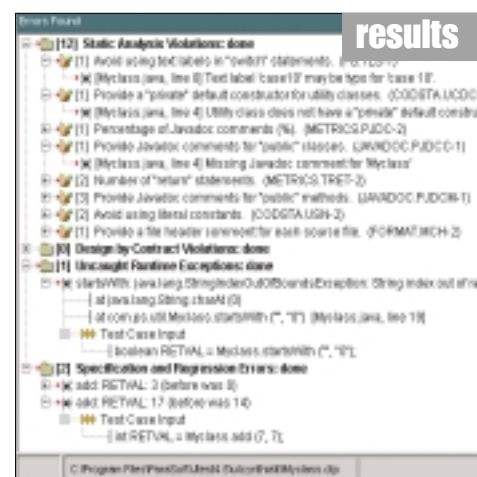
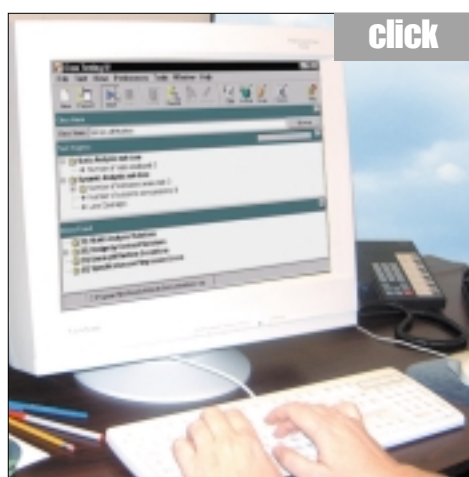
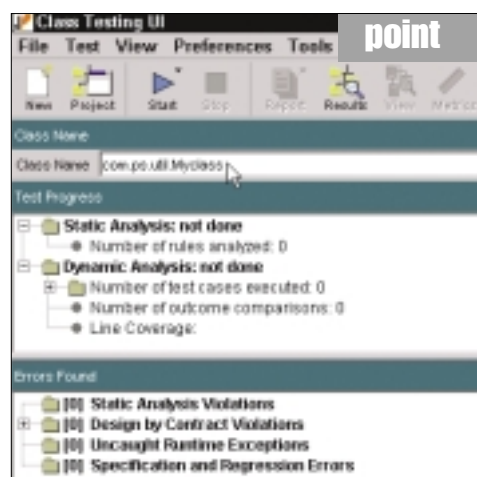
2nd Runner-Up: Oracle JI SCM

Oracle Corporation www.oracle.com

3rd Runner-Up: AllFusion Harvest Change Manager

Computer Associates <http://ca.com>

Automate unit test case generation for JUnit and Java™ with Parasoft Jtest.



(It's as easy as 1-2-3.)

Parasoft Jtest is the first and only automated unit testing tool for Java™ development.

With just a click, Jtest reads and analyzes code — quickly creating harnesses, stubs and test inputs — and tests without user intervention. Jtest also enables you to automate regression testing and static analysis. For loyal JUnit users, Jtest is designed to fully support existing test cases and automate the creation of new JUnit-compatible test cases.

Learn how Jtest can enhance JUnit capabilities...

Download a free eval copy of Jtest along with our informative new white paper entitled "Using Jtest With JUnit."

•----- For Downloads go to www.parasoft.com/jdj_02. Or call 888-305-0041.

Copyright ©2003 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.

New for Java development...

Parasoft® Jtest® 5.0

Features:

- Fully integrated into JUnit & Eclipse
- Quick Fix enabled finds & fixes errors fast
- Intuitively designed GUI

Platforms:

Linux
Solaris
Windows 2000/XP

A part of Parasoft Automated
Error Prevention (AEP) Solutions and Services



PARASOFT
jtest®

**BEST WIRELESS APPLICATION****Winner:** *IBM WebSphere Everyplace Access**IBM* www.ibm.com

WebSphere Everyplace Access version 4.2 simplifies wireless for the enterprise. It delivers the technology that gives mobile employees access to productivity data and enterprise applications from virtually anywhere, at any time. It supports multiple pervasive devices such as PDAs and smartphones from a single platform and helps move business-critical information throughout the organization.

1st Runner-Up: *Advantage Plex and Websydian Computer Associates and Soft Design* <http://ca.com>
2nd Runner-Up: *Oracle9i Application Server Mobile Studio Oracle Corporation* www.oracle.com
3rd Runner-Up: *Wapaka Microbrowser Digital Airways* www.digitalairways.com

**MOST INNOVATIVE JAVA PRODUCT****Winner:** *IntelliJ IDEA**JetBrains, Inc.* www.intellij.com

IntelliJ IDEA is a Java IDE that's filled with development features such as industry setting refactoring support; intelligent code editing assistance; a wide range of J2EE development features for rapid Web application and other enterprise development; a powerful Code Inspection tool; integrated CVS, VSS, and StarTeam support; and an open API for third-party plug-in support.

1st Runner-Up: *Fiorano Tifosi 2002 Fiorano Software, Inc.* www.fiorano.com
2nd Runner-Up: *Eclipse Eclipse.org* www.eclipse.org
3rd Runner-Up: *WebSphere Application Server Enterprise IBM* www.ibm.com

**BEST JAVA REPORTING TOOL****Winner:** *Style Report Enterprise Edition**InetSoft Technology Corp.* www.inetsoft.com

Designed to bridge the gap between enterprise reporting and business intelligence, Style Report/EE is a reporting/BI component that can be embedded inside business processes. The base of Style Report is an enterprise reporting engine with all the mature enterprise reporting features businesses rely on.

1st Runner-Up: *Oracle9iAS Reports and Discoverer Oracle Corporation* www.oracle.com
2nd Runner-Up: *Crystal Reports Crystal Decisions* www.crystaldecisions.com
3rd Runner-Up: *JFreeReport The Object Refinery* www.object-refinery.com

**BEST JAVA APPLICATION MONITORING TOOL****Winner:** *JDBInsight**JInspired* www.jinspired.com

JDBInsight brings developers, testers, and DBAs together. Its performance and transaction information visualizations are tailored for tuning and testing J2EE applications.

1st Runner-Up: *Quest PerformaSure Quest Software, Inc.* www.quest.com
2nd Runner-Up: *Dirig PathFinder Dirig Software* www.dirig.com
3rd Runner-Up: *IronEye SQL from IronEye IronGrid* www.irongrid.com

**BEST JAVA PRODUCER PLATFORM****Winner:** *IBM WebSphere Application Server v5.0**IBM* www.ibm.com

IBM's WebSphere Application Server v5 is a J2EE and Web services technology-based application platform, offering production-ready application servers for the deployment of enterprise Web services solutions for dynamic e-business. It provides the standards-based infrastructure to integrate business processes across the enterprise and with partners, suppliers, and customers.

1st Runner-Up: *BEA WebLogic Workshop BEA Systems* www.bea.com
2nd Runner-Up: *Sun ONE Web Server Sun Microsystems* www.sun.com
3rd Runner-Up: *Compuware Optima! Compuware Corporation* www.compuware.com

**BEST J2EE APPLICATION DEVELOPMENT FRAMEWORK****Winner:** *BEA WebLogic Workshop**BEA Systems* www.bea.com

BEA WebLogic Workshop is an integrated development framework that enables application developers to create, test, and deploy enterprise-class Web service applications on the BEA WebLogic Platform. It provides a unified development platform that enables developers to build and connect components, data, and application business logic, while insulating them from the complexities of J2EE.

1st Runner-Up: *IBM WebSphere Studio (Application Developer v5.0) IBM* www.ibm.com
2nd Runner-Up: *Oracle9i JDeveloper - BC4J Oracle Corporation* www.oracle.com
3rd Runner-Up: *Advantage Gen and Advantage Joe Computer Associates* <http://ca.com>

**BEST JAVA CLASS LIBRARY****Winner:** *WebObjects 5.2**Apple* www.webobjects.com

WebObjects provides a solution when complex data sources are involved and high complexity is required in application logic. It fosters reduced coding and supports a more pure MVC application where no code is needed in the Viewers. WebObjects applications can be deployed as standard J2EE servlets, and it reduces the complexity of developing a J2EE application.

1st Runner-Up: *Java Object Cache Oracle Corporation* www.oracle.com
2nd Runner-Up: *JFreeChart The Object Refinery* www.object-refinery.com
3rd Runner-Up: *Quest JClass Quest Software, Inc.* www.quest.com



Improved Java Code Generation
NEW Enhanced Database Connectivity
XML Differencing
XPath 2.0 Analyzer

improved XML Editor
XSLT Debugger and WSDL Editor

XML

Schema

XSLT

WSDL

Build advanced XML and Web services applications with Altova's **xmlspy**® 2004, the industry-leading XML Development Environment! Minimize the potential for Java errors by utilizing the powerful template-driven code generator that automatically generates Java class files from XML Schema models. **xmlspy**® is the right tool for any XML development task, and the ultimate productivity enhancer for J2EE developers.

Download **xmlspy**® 2004 today: www.altova.com

ALTOVA®

www.altova.com

Provisioning and Digital Rights Management

Instrumentation of J2ME applications

by Shane Isbell

The mobile industry is standardizing both application provisioning and digital rights management (DRM). Provisioning includes the discovery and downloading of an application to a client device, while DRM deals with how to protect the application from unauthorized use.

The Java Community Process recently finalized JSR 124, J2EE Client Provisioning Specification, which a number of content server providers are rapidly incorporating into their products. On the DRM side, the Open Mobile Alliance has defined the industry standard in its document Digital Rights Management version 1.0. Nokia, a strong supporter of the OMA, implemented a key OMA DRM feature – forward locking – in its 3595 model. Forward locking occurs when the device embeds a hardware identifier in the application so that the application can be used only on that device.

One goal of this article is to demonstrate how to use class-level instrumentation to provide DRM solutions for J2ME applications. I'll show how to create a J2ME DRM wrapper that's similar to forward locking, but less restrictive. Another goal of this article is to provide insight into how provisioning and stocking (the submittal and registration of content) takes place.

Software Requirements

Running the sample application requires a Web or application server with a servlet container. JBoss is a popular open source product that serves this purpose. For the client side, download Sun's Wireless Toolkit for MIDP 2.0 from <http://java.sun.com>. The toolkit also contains the sample JAR and JAD files used in this article. If you don't have a J2ME wireless mobile device, use the toolkit's emulator.

You'll need the source code that's included with this article to build the provisioning WAR file and the client content submittal program (download from www.sys-con.com/java/sourcec.cfm). Ant scripts are included; to use them, download the Ant build utility at <http://apache.org>.

Stocking Content

Submitting Content from the Provider

To submit content, the content provider places the JAR file that contains the J2ME application on a publicly accessible URL. Next, the provider creates a JAD file containing seven

required properties, including the MIDlet-Jar-URL property. Finally, the content submitter posts the JAD file to the content server.

It's common in the industry for the content provider to submit both the JAR and the JAD files to the content server. However, since the JAD contains the JAR's URL, submitting the JAR file is unnecessary. Note that the JAD file contains the MIDlet-Jar-URL property so that the application manager on the mobile device knows where to find the J2ME application. We'll use the MIDlet-Jar-URL property in a different way: to get the JAR file from the provider's server onto the content server.

Let's see how to make this work in practice. Go to the WTK20/apps/games/bin directory. There are two files: `games.jad` and `games.jar`. Make the `games.jar` accessible on a Web server and test that the JAR exists by typing in the URL on a Web browser at <http://localhost:8080/ROOT/games.jar>. If a dialog box appears asking to download the file, then the JAR file is accessible. Note that you should replace `localhost` and `8080` with the domain name and port of your Web server.

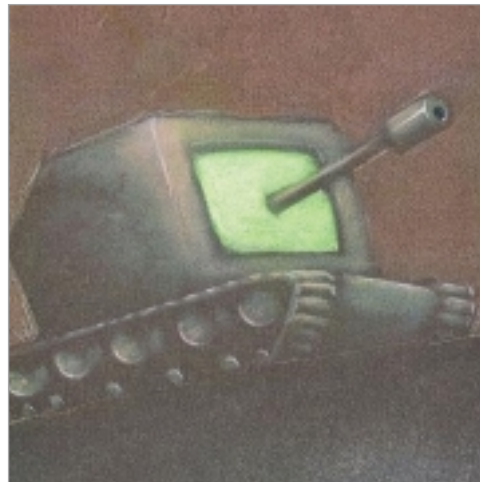
Open the `games.jad` file and change the MIDlet-Jar-URL property to `http://localhost:8080/ROOT/games.jar`. Also add an Install-Notify property with a value of `http://localhost:8080/provisioning/InstallNotify`. I'll explain why you need the Install-Notify property in the section on OTA provisioning. Now instantiate the `org.jvending.vending.client.ContentSubmitter` class, feeding in two arguments on the command line: the URL of the content server stocking servlet

and the local file system path of the `games.jad` file. The instance of the `ContentSubmitter` class will post the contents of the `games.jad` file to the content server over HTTP.

Stocking Content on the Server

Look at the `Stocker` class provided in the source code to understand the stocking process from the content server's perspective. An instance of this class divides the stocking of content into the following steps:

1. Reading the request input stream and creating a JAD object
2. Getting the MIDlet-Jar-URL property from an instance of the JAD object and pulling the JAR file located at that URL to the server



Shane Isbell works as a software architect at a wireless carrier.

random7@att.net

Get the complete picture



Features, Performance and Control

Discover the ILOG JViews Graphics Components

You're developing a sophisticated user interface for a desktop, applet or servlet application – it needs to provide displays that go far beyond what Swing and HTML offer. How can you be sure it will have the features, performance, customization and scalability to enable your end-users to make better more informed decisions, faster?

With ILOG JViews, you get comprehensive graphical libraries & tools, resources, and maintenance services so you can focus on the implementation, confidently completing your application in less time and at less cost.

Quickly and easily build:

- Gantt and resource displays
- Graph layouts, diagrams, workflows
- Geographic map displays
- Realtime data charts
- Custom monitoring and control screens
- Network and equipment management screens

**Get a JViews Info Kit – Learn more, test drive an Eval.
Go to: jviews-info-kit.ilog.com or Call: 1-800-for-ILOG**



Changing the rules of business™

3. Adding the DRM wrapper to the JAR file and setting the new JAR size (MIDlet-Jar-Size) on the JAD instance
4. Adding the modified JAR file to the local in-memory cache

The first step in the stocking process begins after the content submitter posts the JAD file. The content server places the JAD properties within an instance of the JAD class. The JAD class contains accessor and mutator methods for each of the seven required properties, as well as some additional methods, such as the `getMIDlets` method, which returns a map of the MIDlet names. It's important to store this information because the content server needs to know each MIDlet class so it can instrument the class with a DRM wrapper prior to the download of the J2ME application to the mobile device.

In the second step, the `Stocker` object gets the JAR file containing the J2ME application from the content provider's server. The `Stocker` object creates an instance of `JarFetcher` and invokes the `fetch` method, using the JAD object as a parameter. The `JarFetcher` opens a `JarURLConnection` to the JAR URL specified within the JAD and downloads the `games.jar` file from the content provider's server. The `fetch` method returns a `java.util.jar.JarFile` instance. Since the content server instruments the main MIDlets class files, we need to pull out specific class files. Thus we prefer using a `JarFile` object over the lower-level `InputStream` because the `JarFile` class has handy methods for accessing each class file within a JAR file.

In the third step, the content server instruments the class files. The `Stocker` object instantiates the `DrumInstrumenter` class and invokes the `modifyJar(Jad jad, JarFile jar)` method. This method invokes the `getMIDletNames` method on the JAD object to determine the MIDlet class names. Next, the `modifyJar` method pulls the byte code of the MIDlet classes from the JAR file object. The `DrumInstrumenter` instance then instruments the MIDlet classes, adds the DRM class (`ClientAuthenticator`) to the `JarFile` object, and returns the new JAR as a byte array.

The `Stocker` object expands the JAR file size when it instruments the class files. The Application Management Software (AMS) on the device is responsible for downloading and installing the J2ME application. If we don't reset the JAR size, when the AMS downloads the application it will return an error because the JAR size given in the JAD won't match the actual downloaded JAR size. Thus, the `Stocker` object resets the JAR size attribute by invoking `jad.setJarSize(String.valueOf(drmJar.length))`, where `drmJar` is the JAR byte array.

The final step in the stocking process involves putting the JAD object and the instrumented JAR byte array into separate `HashMaps` indexed with the same universally unique identifier (UUID). The UUID is important for a couple of reasons. First, when the user downloads the JAD and JAR, the content server uses the UUID to return each object from the cache. Within a production environment the content server would, of course, persist the JAD and JAR to a file system or database.

The second reason the UUID is important is that the content server will use the unique ID for authorization. The content server embeds the application UUID within the application. When the user initializes the J2ME application, the application will post the UUID to the content server. The server then matches the user's Mobile Subscriber ISDN (MSISDN) in the HTTP header to the UUID of the application to determine authorization.

DRM Instrumentation Wrapper

There are a couple of ways we can create a J2ME wrapper. We can either add all of the code directly to the MIDlet subclass, or we can create a separate class that contains the DRM code and reference that class from the MIDlet subclass. For this implementation we'll choose the second approach because processing time is faster, instrumentation is easier to code, and, most important, we avoid having to deal with the pesky stack map attribute that is unique to J2ME classes. The stack map improves the efficiency of the runtime verification of J2ME applications during runtime by recording local variables and stack items for byte-code offsets.

Note that to avoid the stack map attribute within our instrumentation, we must not include conditional if statements and try/catch blocks within the instrumented byte code. Therefore, the MIDlet subclass has only one public method (`run`) with a void return type. See the CLDC Spec 1.0 for more information about the stack map attribute.

Take a look at the DRM class, `org.jvending.wrapper.ClientAuthenticator`. This class contains only two public methods: `ClientAuthenticator(MIDlet MIDlet, String id)` and `run()`. An instance of the MIDlet subclass, e.g., `TilePuzzle`, invokes the constructor of the `ClientAuthenticator`, passing in a reference to itself (`TilePuzzle`) and the unique ID of the game as parameters. Next, `TilePuzzle` invokes the `run` method. View the code below:

```
String id = "abcd-efa3-sddaf-467sdk";
ClientAuthenticator ma = new ClientAuthenticator(this, id);
ma.run();
startNewApp();
```

The DRM wrapper should make an authorization call to the server upon startup of the application. Thus we embed the authorization code prior to the initial application logic within the `startApp` method. This requires instantiating the `ClientAuthenticator` class at the beginning of the `startApp` method. If the `startApp` method already contains stack map attributes, we need to change the byte offsets and references to the constant pool because we're adding byte code to the beginning of the method.

To avoid dealing with the stack map attribute, rename the `startApp` method to `newStartApp` method. This keeps the offsets and constant pool references the same. The `startApp` method in the MIDlet now invokes the `newStartApp` method,

“Digital rights management and content provisioning are two critical components of the rapid growth of applications in the mobile space”

which contains a copy of the original execution code for the MIDlet subclass.

OTA Provisioning of the Application

A critical step to providing a DRM solution is to authenticate the user. Otherwise the user could forge the HTTP header and claim to be someone he or she is not, thus getting unauthorized access to the application. When users use their mobile device within a GPRS carrier environment, the request goes through a base station subsystem, through the serving GPRS support node (SGSN), onto the gateway GPRS support node (GGSN), and finally to the WAP gateway.

By this point, the system has already authenticated the user and appended a MSISDN, which is the user's phone number, to the HTTP header. Thus, by the time the HTTP request hits the content server, the server only needs to extract the MSISDN HTTP header to know the identity of the user. Before going further with the authentication and authorization of the user, let's briefly discuss how the user discovers the instrumented application.

The user can do application discovery through a WML microbrowser or an HTML browser, depending on the capabilities of the handset. After users discover the link to the JAD file, they click it. The URL will look something like <http://localhost:8080/provisioning/596162646162614A787.jad>.

On the content server, any URL with a *.jad extension maps to the `org.jvending.provisioning.JadDownloader` servlet. The `JadDownloader` instance extracts the UUID and gets the JAD from the cache. Next, the `JadDownloader` sets the content type on the response to `text/vnd.sun.j2me.app-descriptor` and returns the JAD stream to the browser.

The device starts downloading the JAD file over WAP. The browser detects that there is a content type of `text/vnd.sun.j2me.app-descriptor` and passes control over to the AMS. The AMS on the device reads the properties of the JAD file and extracts the `MIDlet-Jar-URL` property, which looks like <http://localhost:8080/provisioning/596162646162614A787.jar>.

The AMS hits this link over WAP or directly over TCP/IP, depending on the device. Since the URL contains a *.jar extension, it maps to the `org.jvending.provisioning.JarDownloader` servlet. The `JarDownloader` servlet extracts the UUID and looks up the instrumented JAR file from the cache. Next, the servlet sets the response content type to `application/java-archive` and downloads the JAR to the device.

The device knows where to post the status report by reading the `MIDlet-Install-Notify` attribute in the JAD. Remember, you should have added this to the `games.jad` file earlier. If the device successfully downloads and installs the application, it posts a 900 response code to the content server at <http://localhost:8080/provisioning/InstallNotify?uuid=596162646162614A787>.

When the HTTP post hits the `InstallNotify` servlet with a successful 900 response, the content server extracts the MSISDN from the HTTP header and the UUID from the URL. The server caches the MSISDN and UUID into an instance of `AuthorizationMap` using the MSISDN as the key.

After successfully downloading the application, the user clicks the application start button on the device. At this point, control passes to the `startApp` method of the main MIDlet. The `startApp` method instantiates the `ClientAuthenticator`

class, which posts the UUID to the content server's `AuthorizationServlet`. The `AuthorizationServlet` extracts the MSISDN and UUID from the header. Using the MSISDN from the HTTP header as a key, the servlet looks up the UUID from an instance of `AuthorizationMap`.

If the UUID in the header doesn't match the UUID in the `AuthorizationMap` instance, the servlet sends an unlock response value of `false`. The `ClientAuthenticator` invokes the `notifyDestroyed` method on the MIDlet subclass instance, ending the application. This need to destroy the application from within the wrapper is why the `ClientAuthenticator` constructor signature contains a reference to the invoking MIDlet subclass.

If the UUIDs match, the servlet sends back an unlock response value of `true` to the J2ME application. The run method returns immediately, authorizing the user. In the MIDlet, control now goes to the `newStartApp` method, which contains the original MIDlet code. The application begins.

Conclusion

Digital rights management and content provisioning are two critical components of the rapid growth of applications in the mobile space. Currently, it's very easy for a user to forward content or to upload it to the Internet for others to download. Not only do we need intelligent ways for the user to discover and download applications, but we also need to protect the content from piracy. This article discusses one way – using a DRM wrapper – to protect content from unauthorized use. The source code in this article is open source. You can find updates at www.jvending.org.

Google Seeks Expert Computer Scientists

Google, the world leader in large-scale information retrieval, is looking for experienced software engineers with superb design and implementation skills and considerable depth and breadth in the areas of high-performance distributed systems, operating systems, data mining, information retrieval, machine learning, and/or related areas. If you have a proven track record based on cutting-edge research and/or large-scale systems development in these areas, we have plenty of challenging projects for you in Mountain View, Santa Monica and New York.

Are you excited about the idea of writing software to process a significant fraction of the world's information in order to make it easily accessible to a significant fraction of the world's population, using one of the world's largest Linux clusters? If so, see <http://www.google.com/cacm>. EOE.



Chemical Early Warning System in Java

Developing emergency response systems

by Kathy Lee Simunich,
Gordon Lurie, Michelle Kehrer,
and Tom Taxon

The need for a chemical/biological early warning system within mass transportation sites is real and was being researched before the September 11 terrorist attacks. Since then, this research has become an operational necessity.

Led by Argonne National Laboratory, along with teams from Sandia National Laboratories (SNL) and Lawrence Livermore National Laboratory (LLNL), researchers created an early warning system for subways within a major metropolitan area. The entire system entailed the processing of sensor data, live meteorological data, video feeds, and real-time track/train data; execution of below- and aboveground dispersion models; and the timely display of results to subway and emergency personnel. The system is based extensively on Java technologies, and this article examines the merits of each J2EE package used.

The Chem-Bio Emergency Management Information System (CB-EMIS) is an enterprise-level distributed system. The decision to write it in Java was made early because of Java's maturity, scalability, and maintainability, as well as the availability of the industry-supported J2EE technologies. This article discusses the software aspects of the CB-EMIS and presents the architecture in detail.

Industry Standard Technologies

The CB-EMIS runs on four dual-processor Dell PCs running the Red Hat 8 Linux Operating System (the servers). Because the system must always be up and able to recover from communication failures, etc., three J2EE technologies were used to gain this reliability: Java Messaging Service (JMS), Remote Method Invocation Object Activation Daemon (RMID), and persistence through Java Database Connectivity (JDBC). A relational database was used to persist data for forensic playback and simulation modes (MySQL for development and training, and Oracle 9i for operational use).

The Sun ONE (previously iPlanet) Message Broker was used for the JMS system, and Novell's Lightweight Directory Access Protocol (LDAP) implementation (eDirectory) was set up for use as the directory service and for the authentication of users. It's important that the server processes across all CPUs remain synchronized in time; a Network Time Protocol (NTP) server was used to keep the system times in sync. Since the system relied on JMS messages, it was important to have the times match for message time-out purposes. Also, the Time Server process reads the system time to keep a consistent time stamp for all data and messages passing through the other server processes.

CB-EMIS System-Level Services

This section describes the design of the foundational services layer with which all server processes extend, implement, or interface. Figure 1 shows the services layer that supports the CB-EMIS.

Directory Service

The LDAP server was used for authentication of user accounts as well as a directory service for object binding. The schema was also extended to allow the storage of attributes unique to the system on a per-user basis, allowing control over what each user could see or do in the client graphical user interface (GUI). The Java Naming and Directory Interface (JNDI) package was used throughout the CB-EMIS to bind and look up object references stored in the LDAP server. Figure 2 shows the various objects bound into the directory. Not only were the JMS Topics and Topic Connection Factory references bound in the server, but the various JDBC data sources were as well. In this way, any of the CB-EMIS processes can look up the references to all the connection factories, topics, and JDBC data sources without having provider-specific references within the code. Saving these objects within an LDAP server gains persistence and security, and avoids the need for a shared file system. The team also decided to bind the CB-EMIS server process Remote Method Invocation (RMI) stub references within the LDAP server. Because there's not a JNDI class for binding stub references within a directory server, it was necessary to write our own `javax.naming.spi.ObjectFactory` and `javax.naming.Referenceable` classes to serialize/deserialize the `java.rmi.server.RemoteStubs`.

The Archive

A relational database management system (RDBMS) is used as the persistent store (referred to as the Archive) for the static and dynamic data during either a simulation or normal operation. The purpose of persistence is for error recovery and playback mode. Static system description data is imported into the database for use during normal operation or simulations. During playback mode of either a training simulation or a normal operational session, the data archived in the database is retrieved and executed for the user. Archived live data can be played back for forensic study of an incident.

Each server process looks up the data source via JNDI and connects to the appropriate database through the JDBC. All events and data are saved in the database for playback or for error recovery. Rather than having SQL statements scattered throughout the code, a Persistent Object class was created that other classes extend. By defining the table name and the column names, each class inherits the ability to insert, retrieve, update, and delete itself from the database. JDBC Prepared Statements were used to speed up transactions to the database.

A Persistence Broker class does the JNDI look-up, maintains the reference to the data source, and automatically reconnects to the database if the connection is lost. Most of the data records are time-stamped, which is usually part of the primary key.



Forget something?

Post-launch is NOT the time to be verifying web applications.

The wild blue yonder of operational monitoring and management is extremely unforgiving. Which means that going live with the monitoring software you used in development is a great way to go dead—quickly! You simply can't support operations if your staff is drowning in details provided by development profiling tools and debuggers. **Let NetIQ cover your apps...with AppManager.**

AppManager—the industry's easiest-to-use Systems Management suite—is a proven management system for monitoring J2EE application servers, databases, operating systems and even end-user response time. NetIQ's AppManager monitors ALL application components—not just your server. **NetIQ. Nobody does UNIX better. Nobody.**

Visit us at www.netiq.com/solutions/web to learn how we can help you address the challenges of your operational monitoring and management.



Activatable Server Processes

Each of the server processes is restartable via RMI activation. Activation allows programs to register information about remote objects so they can be started up when they are first accessed or when the RMI Daemon (rmid) starts up. This provides the ability to start the entire server, which consists of processes across multiple machines by registering one process to be started when rmid starts up, as well as the ability to recover from a Java Virtual Machine (JVM) failure by restarting a failed process. If the JVM of an activatable object does not exist, the RMI Daemon creates a new JVM for that object.

All of the CB-EMIS server processes are restartable by extending the `java.rmi.activation.Activatable` class. The process stubs are registered with the rmid process (which is already running) and also binds its stub reference into the LDAP server. The RMI Daemon provides a Java Virtual Machine from which other JVM instances may be spawned. This process actually starts up the other server processes “on-

demand” as they are requested. Therefore, if a process goes down it will be reactivated upon the next reference to it via rmid (barring an actual hardware failure).

A stub reference is looked up by a CB-EMIS process via JNDI and passes that reference to the RMI Daemon, which in turn activates the process.

JMS Messaging

All the server processes communicate via a publish/subscribe methodology with JMS Topics. This communication mechanism was chosen because JMS has several advantages over Remote Method Invocation for this type of application, most notably:

- It is asynchronous in nature, so if one component goes down, the process that’s sending the data is not tied up waiting for the other process to restart.
- Messages can be guaranteed to be delivered even if the process that needs it is not running. It will get all waiting messages when it starts back up.
- It’s fairly simple to have another process register to receive data from the same source as others without having to contact the source of the data.

The server processes look up references to the JMS Topics that they’re interested in and “sign up” as a publisher and/or a subscriber to various topics. Like the database connections, the JMS Utility class tries to reconnect if a connection is lost or dropped. Most classes in the CB-EMIS that are persistent usually also implement the Publishable interface. Instead of making RMI calls to pass data or having the processes make JDBC calls to pass data via a database, the data is wrapped up in a JMS message and published to a topic.

CB-EMIS Server Processes

Figure 1 shows the CB-EMIS Server Processes interacting with the services layer. The CB-EMIS “server” has been broken down into the various components (or processes) described in this article based on the various tasks the server needs to do.

The processes fall into the following categories: data collection, monitoring, and utilities. The data collection processes are the Meteorological Data Collector, Live Met Data Feed, Track Data Collector, Sensor Data Collector, and the Critical Actions Collector. The monitoring processes are the Alarm Generator, System State Updater, and System Monitor. The utility processes are the Time Server and the Model Executor.

Thus, the server consists of 10 separate processes running within their own JVM across the four dual-processor PCs. They all are activatable via the RMI Daemon. When they start up, they make a JNDI look-up to get references to the JDBC data source and make a connection to the database as well as to the JMS Broker to sign up for publishing and/or subscribing to the available topics. All of the processes employ multithreading to provide the fastest possible processing of information through the system, while monitoring all the data topics and archiving to the database.

This section describes each server process, its basic purpose, and the relevant Java technology it employs to do its work. As stated earlier, the processes communicate via JMS Topics. These include:

- **System Control:** The System Monitor publishes to this topic whenever the administrator changes the mode of the server (normal, playback, or training). All processes subscribe to it and change their processing according to the current mode of operation.
- **Process Status:** The System Monitor also publishes to this topic approximately every two seconds so that the administrator GUI can show the current state of the server processes.

Kathy Lee Simunich has been an object-oriented software engineer for over 15 years, progressing from C++ to Smalltalk and now writing almost exclusively in Java since 1998. Her concentration has been in cross-platform model integration and simulation systems. She has a BS in meteorology and an MS in computer science.

Gordon Lurie has been an object-oriented architect and software engineer for over 15 years. He was the software architect lead developer for the CB-EMIS system. His work has focused on object-oriented geographic information systems. He has a BS in computer engineering.

Michelle Kehrer has been a software engineer for over 10 years developing standalone client, Web, and enterprise applications. She has a BS in computer science.

Tom Taxon has been an object-oriented software engineer for over 13 years. He worked with Smalltalk and C++ and has been working in Java since 1996. His background is in GIS and model integration and simulation systems. He has a BS in computer science and engineering.

partners@anl.gov

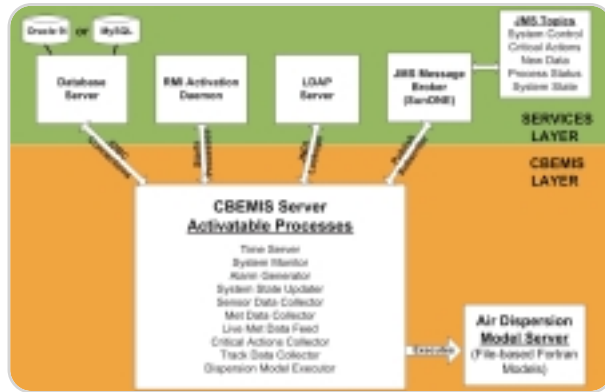


Figure 1 CB-EMIS Software Elements

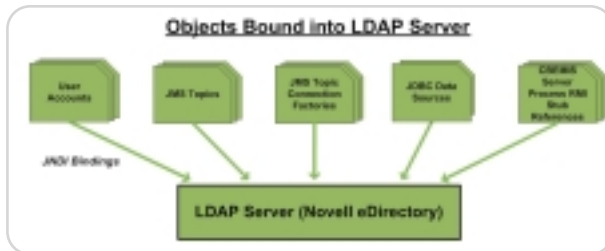


Figure 2 Objects bound into LDAP server

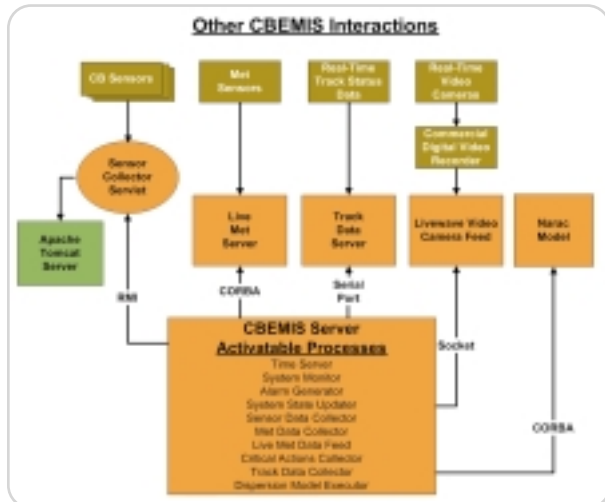


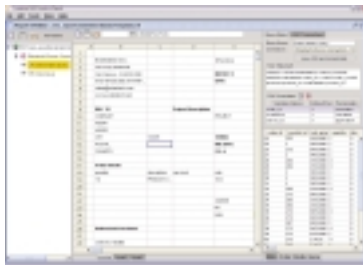
Figure 3 CB-EMIS interactions with other hardware/software elements



Houston,
we've found a solution...

Apollo 17 Astronaut driving Lunar Roving Vehicle (LRV). December 1972

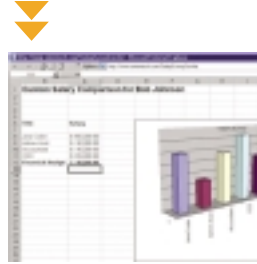
ExtenBIS: Know your business, accomplish your mission.™



ExtenBIS Data Mapper



Excel Spreadsheets



ExtenBIS Business Intelligence Server

Extentech understands that your users need their data presented with all of the capabilities of an Excel spreadsheet, but don't have the bandwidth to learn yet another report design tool. Well, what better tool to create advanced spreadsheet reports with than Excel itself?

With the **ExtenBIS Data Mapping** and **Deployment Wizard**, you can visually map your real-time database, session, and bean values to new or existing Excel templates with a drag-and-drop interface, then deploy it over the web to any server running the **ExtenBIS Servlet** with a single mouse-click. By creating server-based Business Intelligence applications from these files, ExtenBIS is the key to achieving the maximum ROI from your existing BI content.

Utilizing a fully-customizable JSP-based portal (source-code included,) the ExtenBIS Business Intelligence Server delivers real-time Excel reports to your users anytime, anywhere.

EXTENBIS
BUSINESS INTELLIGENCE SERVER

- ✓ Easy to use 3-Step Report Mapper and Deployment Wizard
- ✓ Secure Delivery of Reports via Web, Email, and File Drop
- ✓ Customizable Business Intelligence Portal with Source Code
- ✓ Customizable Business Intelligence Servlet with Source Code
- ✓ Extensible Object-Relational CellBinder API with Source Code
- ✓ XML Report Definition Format

Download ExtenBIS: www.extentech.com/jdj/

extentech™

- **New Data:** All server processes archive data and events to the database and then publish the data on the New Data topic for the other processes to ingest (if interested in the particular data item).
- **Critical Actions:** The CB-EMIS client process publishes to this topic whenever the emergency response users declare a critical action (e.g., trains have stopped, alarms triggered). The purpose of this topic is to provide a non-RMI method of having the client processes communicate critical information back to the server processes.
- **System State:** The System State Updater publishes eight different data types to this topic. Each data type runs in its own thread, publishing frequently enough to update GUI elements within the client process. By publishing the state of the system onto this topic, any number of client processes can log in and display the current state by subscribing to the topic. As a consequence, clients can also recover from communications interruptions without missing data.

The Time Server

The Time Server is responsible for holding and maintaining the system time. It's the central point for all the server processes to obtain the synchronized system time, since it is the process that is synchronized with the NTP server. The other server processes access the time from the Time Server process via an RMI call. Having a single process keep track of the system time allows all processes to use the same methodology for getting current time regardless of the current mode (normal, training, or playback).

The System Monitor

The System Monitor is responsible for monitoring the state of the software system. It restarts the components if there is a

problem and notifies the user if it cannot restart a process. The status of the system processes (components) is displayed via a GUI for the system administrator.

The System Monitor is responsible for verifying that all required processes are running and maintaining the current run state (mode, simulation run status, simulation speed, etc.). It pings a process via RMI every two seconds. When a process is pinged, the activation system (RMI Daemon) starts up the process if it is not already running. The individual processes must also ping the system monitor back within five seconds. This allows the administrator to be notified if a process starts up and then crashes immediately. In addition, if a process detects or recovers from a nonfatal error, it can notify the System Monitor of this state change via an RMI call. If the status of a process changes, the current status is published on the Process Status Topic.

The current run state is maintained via RMI calls from the System Monitor GUI. When changes are made to the state, they are published to the System Control Topic so that all other interested processes can handle the change.

System State Updater

The System State Updater is responsible for maintaining the system state (train location, track status, meteorological conditions, sensors, plumes, alarms, etc.) and publishing it periodically to the System State Topic. There is currently one updater for high-bandwidth connections to the server, as there is a full, detailed system state sent every second. It listens to the New Data topic for when any process saves data to the database and then publishes the new data on the appropriate System State Topic for consumption by client processes that are currently connected. The design allows for other levels of detail of the System State to be published. One

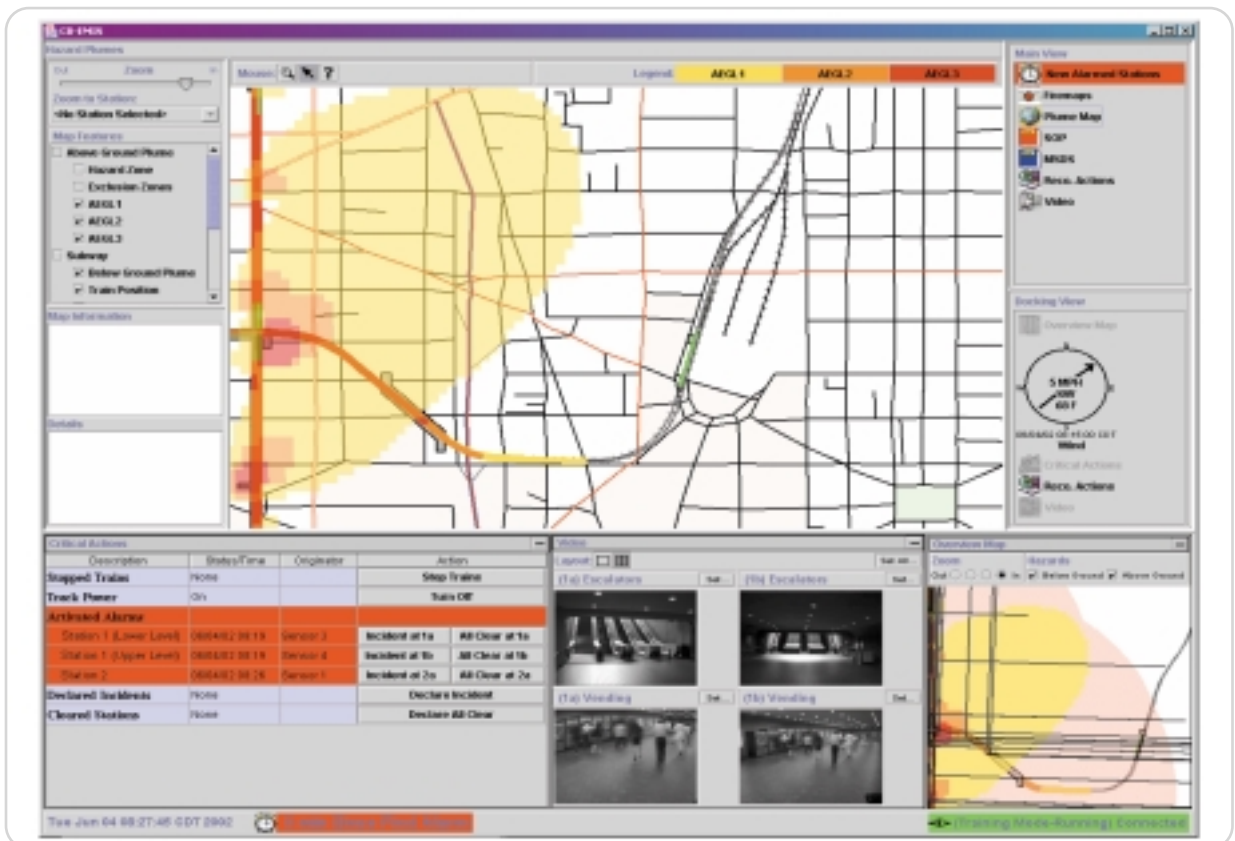


Figure 4 CB-EMIS Client GUI main window showing plume dispersion map with contamination, critical actions table, and various video feeds

such use of this would be a handheld computer used by an emergency responder; it might get only system state data relative to the responder's location. Thus, a low-bandwidth updater can be created that publishes updates less frequently for clients connecting over a dialup or wireless connection.

Alarm Generator

The Alarm Generator is responsible for monitoring the New Data Topic for sensor events, determining if an alarm condition has been reached, and storing the data in the Archive/DB. It also publishes any alarms to the New Data Topic.

Model Executor

This process is responsible for responding to sensor alarms in the New Data Topic. It generates input files based on data received from the New Data Topic and starts to execute the appropriate below- and aboveground dispersion model. When the model(s) are finished, this process publishes the contamination results to the New Data Topic and archives the data in the database.

Critical Actions Collector

The main purpose of this process is to provide a non-RMI method of having the client process(es) communicate with the server. This is essentially a pass-through for Critical Action Data messages. It tracks who created a critical action and when it was generated.

Sensor Data Collector

This process is responsible for monitoring all the chemical/biological agent sensor readings. It stores the readings in the Archive and publishes the data on the New Data Topic. This process registers with the SNL Sensor Server via RMI.

Live Met Data Feed

This process polls the LLNL Met Server, which monitors real-time weather observations, and stores the data in a local database for access by the Meteorological Data Collector.

Meteorological Data Collector

The Met Data Collector is responsible for retrieving meteorological data from a server (or from a file if in training mode), storing the data in the Archive/DB, and publishing to the New Data Topic. This data is needed for the dispersion models to correctly predict plume concentrations.

Track Data Collector

The Track Data Collector is responsible for receiving real-time track status data and train location data, archiving it in the database, and publishing the data on the New Data Topic. It subscribes to the Critical Actions Topic for notification if the trains stop running.

CB-EMIS Interactions with Other Hardware/Software

Figure 3 shows the CB-EMIS interactions with various servers that have the physical connections to the sensors as well as other parties' monitoring software with which the CB-EMIS must communicate. Four types of sensor hardware are needed for the system:

1. Chemical/biological agent detectors
2. Meteorological sensors
3. Real-time train location and track status sensors
4. Real-time video camera feeds

Each of these systems has associated servers that were installed by parties other than Argonne. The video cameras were part of a legacy system installed for security purposes that CB-EMIS tapped into for an integrated display within the client GUI. The train location and track

Schmo.

Pro.

Gets job advice from the newspaper.

Goes to Dice.com first for complete tech career guidance.

Dice[™]
Look to the tech leader first.[™]

Only tech jobs • Career guidance • Salary info • Daily job notification • Training & certification

© 2004 Dice Inc.

status data feed were also part of a legacy system used by other monitoring software within the subway's control room. This data, along with current meteorological conditions and sensor readings, contains the critical information needed for accurate predictions of agent plume dispersion and concentrations.

Argonne has written a belowground dispersion model specifically for the subway where the CB-EMIS has been installed. The system also employed an existing EPA aboveground model (INPUFF) for the prediction of plumes if the agent escapes from underground. These models are written in Fortran and therefore use flat files for input and output. The CB-EMIS Model Executor reads/writes the output/input and makes a system call to actually start the models running on a separate CPU.

The CB-EMIS also uses the LLNL real-time meteorological observation server. Data is updated every 15 minutes, and the Live Met Data Feed process polls the server for current observations. The Model Executor can also access and run LLNL's high-resolution aboveground dispersion model (Narac) at a preset time after an alarm to obtain more accurate results. The default INPUFF model is useful for quick assessments, but more detailed predictions can be calculated with the Narac model (although it takes longer to run and get the results).

Sandia was responsible for installing the sensor detection software interface to log the actual readings of the chemical/ biological agent detectors. They implemented their sensor collection software as a Java servlet and, therefore, the servlet must run within a servlet container. The Apache Tomcat Server is used as the servlet container and Web server and runs on one of the nodes in the Linux cluster. The servlet continuously monitors the sensors.

CB-EMIS Client

Figure 4 shows one view of the Swing-based GUI for the emergency response coordinators. Each panel on the display can be either minimized and shown in icon form or maximized to show more detail. Panels include:

- GIS-like overhead map view with zoom and pan capability
- Overview map of subway system and train status
- Plume concentration overlay
- Sensor concentration-level graphs
- Critical actions and alarms
- Video camera views

Any number of remote client processes can be communicating with the CB-EMIS server located in the control center of the subway system. This communication link is done through Secure Socket Layers (SSL) and uses the Java Secure Socket Extension (JSSE) package to make the connection through the Java client process to the proxy server. As mentioned in the JMS Topic section, JMS messaging is used to pass data between the client process and the CB-EMIS server over the SSL channel. Critical actions are posted to the server via the Critical Action Topic, and each client subscribes to the System State Topic for updates. These updates are viewable in near real time on a high-bandwidth connection.

Conclusion

Emergency response systems, such as the Chem/Bio Emergency Management Information System, should be deployed throughout many critical infrastructures such as subway systems, airports, or other mass-transportation hubs. This article presented the distributed processes, the need for asynchronous communication through messaging, and secure communication and user access mechanisms provided by the CB-EMIS employing various Java technologies.

Reliability was achieved by using the RMI Activation Daemon to reactivate any processes that may have failed, and persistence

of JDBC Data Sources and JMS Topic Factories within an LDAP server allowed restarted processes to easily look up the references via JNDI and reconnect to the database and message broker automatically.

Security was maintained through user authentication within the LDAP server on both the CB-EMIS server side as well as access to the client processes. RMI security policy files were used to restrict access rights for any RMI calls throughout the server processes. Further security was achieved by using an SSL connection via JSSE from client processes to a proxy server and then to the server processes behind a firewall.

This project has proven that Java technologies not only addressed the various needs of the system, but helped all the components of the system integrate seamlessly and efficiently. With all the monitoring processes, the continuous real-time updates (not only of the system status, but of the train and track data and sensor readings), the database archiving of all data, and message publishing and subscribing, the performance of the multithreaded processes running in multiple Java Virtual Machines was never a problem.

Q&A

Joe Ottinger, *JDJ's* editor-in-chief, had the opportunity to have a discussion about the Chem-Bio Emergency Management Information System with the team at Argonne National Laboratory.

JDJ: What other technologies were considered? What were the measurements for validity for the technologies being considered?

ANL: We wanted to use the established Java technologies. The first principle of the entire system was reliability.

The early prototype system was a Smalltalk/Java program using RMI and some custom socket-based communication between Smalltalk and Java. The Smalltalk version of GeoViewer was existing code that is now converted to Java for the final product.

Once we converted to production development, we needed to manage the synchronization of the communications between all the server processes and clients, and each process had to be restartable. We chose RMID activation for its restarting ability and JMS for its asynchronous communications and guaranteed delivery of the messages.

These technologies greatly reduced our development time.

JDJ: I notice you're duplicating a lot of the functionality of JMX - how has the JCP worked to affect your development process and architecture (why wasn't it used, in other words)?

ANL: We felt JMX was not a mature enough technology when we started on the final system. We started the final development stage in December 2001 and it had to be delivered/installed by January 2003 and operational by June 2003.

JDJ: What licenses did you use for your custom code, specifically for the RMI stubs in your LDAP server?

ANL: We used LDAP for authentication and the JNDI lookup of the data sources as well as the RMI stub references. The use of eDirectory was dictated by the sponsor.

JDJ: What kind of persistence layers did you consider, if any? Are they in-house, commercial, or open source?

ANL: We had already written an in-house persistence layer for other projects, so we adapted it for use in our final prototype. The database was used more as a logger than needing the full features of an RDBMS. We used generic SQL to switch between using MySQL for development and Oracle for deployment.

JD: You mention you're using iPlanet for some services – why, then, did you choose Tomcat for a servlet container, when iPlanet also has a servlet container available for it?

ANL: We bought just the Sun ONE Java Message Queue since that's all we needed it for. We chose it after testing a couple of other vendors' products. The servlet used to monitor the sensors was a prototype developed by another party that has since been replaced by a more robust system, which is *not* based on servlet technology. It's much more tightly integrated with the rest of the system.

JD: In the "System State Updater" section, what is a "plume"? I have some idea because of context, but clarification would be nice.

ANL: The "plume" is a contour plot of the predicted concentration levels of the released agent that could pose health risks.

The plumes are updated every minute and are plotted as an overlay over the map on the client machines.

JD: What client platforms are supported?

ANL: Windows 2000 and XP have been extensively tested for the client software. However, plans are in the works for supporting Red Hat Linux (which is what we run on the server side). All the code runs on the major OSes, including Solaris, Red Hat Linux, and Mac OS X. Basically, any platform that supports J2SE can support our system. There is a small amount of JNI code included in our system, so the biggest hurdle is just compiling that code for a new platform. Plus, tweaking for optimal bandwidth usage for the video streaming needs to be done on any new platform. ☺

References

- www.anl.gov
- <http://java.sun.com>
- www.dell.com
- www.redhat.com

- www.mysql.com
- www.oracle.com
- www.sun.com/software/products/message_queue/home_message_queue.html
- www.novell.com/products/edirectory/
- www.eecis.udel.edu/~ntp/
- <http://java.sun.com/products/jndi/>
- <http://java.sun.com/products/jdk/rmi/>
- <http://java.sun.com/products/jdbc/>
- <http://java.sun.com/j2se/1.4/docs/guide/rmi/activation.html>
- Coke, L. R., Sanchez, J.G., and Policastro, A.J. "A Model for Dispersion in the Subway Environment." 10th International Symposium on Aerodynamics and Ventilation of Vehicle Tunnels, Principles, Analysis, and Design. Boston, Massachusetts, 1-3 November, 2000
- Petersen, W.B., and Lavdas, L.G. "INPUFF 2.0 A Multiple Source Gaussian Puff Dispersion Algorithm User's Guide." EPA/600/8-86/024, U.S. Environmental Protection Agency, August 1986.
- <http://narac.llnl.gov>
- <http://java.sun.com/products/servlet/>
- <http://jakarta.apache.org>
- Lurie, G.R., Sydelko, P.J., and Taxon, T.N. "An Object-Oriented Geographic Information System Toolkit for Web-Based and Dynamic Decision Analysis Applications." *Journal of Geographic Information and Decision Analysis*. 2002, Vol. 6, No. 2, p.108-116
- Korp, P.A., Lurie, G.R., and Christiansen, J.H. "A Smalltalk Based Extension to Traditional Geographic Information Systems." Proceedings of the ParcPlace/Digitalk International Users Conference, San Jose, CA, ANL/DIS (July 31-Aug. 12, 1995).

THE SUBMITTED MANUSCRIPT HAS BEEN CREATED BY THE UNIVERSITY OF CHICAGO AS OPERATOR OF ARGONNE NATIONAL LABORATORY ("ARGONNE") UNDER CONTRACT NO. W-31-109-ENG-38 WITH THE U.S. DEPARTMENT OF ENERGY. THE U.S. GOVERNMENT RETAINS FOR ITSELF, AND OTHERS ACTING ON ITS BEHALF, A PAID-UP, NONEXCLUSIVE, IRREVOCABLE WORLDWIDE LICENSE IN SAID ARTICLE TO REPRODUCE, PREPARE DERIVATIVE WORKS, DISTRIBUTE COPIES TO THE PUBLIC, AND PERFORM PUBLICLY AND DISPLAY PUBLICLY, BY OR ON BEHALF OF THE GOVERNMENT.

WebCharts3D

One demo worth your download

www.webcharts3d.com/demo



WebCharts 3D is the ideal XML-based charting solution for web applications.

With its powerful charting engine, WebCharts 3D can produce a wide range of different charts in the most demanding enterprise environments.

Deliver charts to browsers and mobile devices as either applets or interactive server-generated images in a variety of popular formats. Or, embed JavaBeans directly into rich client applications.

WebCharts 3D's WYSIWYG approach allows you to begin development immediately after installation.



**Fastest Development.
Fastest Deployment.
Fastest Runtime.**



Neil Poska



Learning to Learn

I have come to the unpleasant realization that I might have forgotten how to learn. I'm not talking about the small bridges we cross every day in our jobs or our lives. I'm talking about the big tectonic shifts that take place far less often.

I suppose it started out when I realized I am jaded. I'm sure there is a difference between "specializing" and "being in a rut," but I can't see it from where I sit right now. I know it's fashionable now to bag on the whole Enterprise JavaBean technology and I don't want to be a part of that, but even that has become more trouble than it's worth to me. As much as I loved the idea of it in the beginning (like most relationships), the emotions have faded as I came to realize how little I was getting out of it compared to the effort I was putting in (like most relationships).

I decided to head off in search of new tools and technologies. I settled upon two of the new players in the Java arena, Spring and Hibernate, because they supposedly solve two issues in every project I have dealt with: Spring provides an MVC framework for me, hopefully covering up

Hibernate has a fantastic documentation set, with many examples and a tremendous reference document that stretches for 138 pdf pages. In addition, there are great user lists and their forum is just about at the perfect activity level. There are enough posts that you can consistently learn, but not so much activity that you have to slog through 500 "I am new to this but..." posts every day. As a result, all my experiments with Hibernate went flawlessly.

Spring, being a somewhat newer product, has some documentation issues. However, in their favor, at least they know it. They often promise that the next version of Spring to be released will completely address the problem; a complete Spring reference manual was due in February and two books are due this year. To counter this issue, the Spring folks enthusiastically support their user forum. One of the Spring contributors even found and responded to my blog, where I was expressing my frustration with them at the time.

Let me take a second to reiterate something: the problems I had with Spring are not entirely Spring's fault.

“It has been many years since I tried to learn a product that was as new as Spring or Hibernate, and it brings up an interesting question: How do you learn these technologies”

my woeful lack of experience in that area; and Hibernate provides a robust object-to-relational mapping tool, allowing me hopefully to completely jettison entity beans from my life.

It has been many years since I tried to learn a product that was as new as Spring or Hibernate, and it brings up an interesting question: How do you learn these technologies? How do you learn anything new? I have been programming Java for so long that if I have a question about a class method or a different way of accomplishing some programmatic goal, I pretty much know where to look.

I know the Javadoc for the Sun APIs to a reasonable level, so finding the missing pieces is pretty easy. I already know where to look and for the most part, what I am looking for. If all else fails, there are hundreds of books available at any bookstore. I can post to an online forum, and people from all over the world will answer (very few of the answers will be helpful, but they will answer :). I can even call 10 of my friends and ask their opinions. Spring and Hibernate are different. They are new, and I feel like I'm on my own.

I started at their respective project Web sites, hoping to start reading all the documentation that was available.

I am in uncharted territory, so I expect some pain. In fact, solving these problems is a fantastic way to learn the technology. Spring is promising, interesting, and useful. But to coin a phrase brought to you by the Department of Redundancy Department, I wouldn't be in uncharted territory if the territory weren't so darned uncharted.

What have I learned from this process? I guess I learned that the learning process never really changes once you have it down. This was, for some reason, very surprising to me. But sure enough, reading documentation (when it exists), reading APIs, and talking to people is still the way to get things done. Beyond that, we learn by doing, and it is the same whether we are trying to learn a new tool or trying to repair whatever is making that hideous noise in the toilet tank.

Most important, I learned how lucky we are to work in a field where we can keep learning and growing. I think about all my friends who aren't programmers; they all have fine jobs, but they will be doing those jobs in mostly the same way until they retire. With a little effort, we can completely overhaul the way we do our jobs. In that way, we never have to worry about being bored or jaded again. ☺

Neil Poska is president of Object One Associates, and has been a software consultant for many years, specializing in distributed object technologies and enterprise applications

nposka@objectone.com



INSTALL-DICTATE-COMMAND

ALL THE COMPONENTS
OF YOUR OWN SOFTWARE,
USING THE NEWEST
SESMA PRODUCTS,
DISTRIBUTE IT, TOO.

JAVA CODE TO DIRECT
THE MS API- ACTIVER,
COM, AND WEB APPS.
JAVA SAYS JUMP,
MS ASKS HOW HIGH.

AN IMPRESSIVE LEAD ON
CODE CONVERSION EFFORTS.
MAKE YOUR MS SOFTWARE
ACT AND RESPOND AS JAVA
AT THE CLICK OF A BUTTON

Lead the March for a Brave New Java Desktop Today

BROWSER

Java API developed for
seamless integration of
Microsoft Internet Explorer
(MSIE) into the
Java Environment.

COM

Java API that allows bi-
directional communication
into COM core services on the
Windows Operating System.

FLASH

Java API for integrating
Macromedia's Flash
technology into the Java
environment without
Internet Explorer as a
heavyweight middle layer.

SESMA is proud to announce **J[®]**.
This summer, creating customized
installers for your applications (and
its Java and MS components) is
simple-quick, clean, and flexible.
Need Proof? We trust all of our
software to it. When you
install SESMA, you're
using **J[®]**.

READER

Java API for integrating
Adobe Acrobat Reader
technology into the Java
environment without
Internet Explorer as a
heavyweight middle layer.

Every piece of SESMA software is part of a bigger picture.

SESMA is also proud to announce the forth-coming messaging system,
jUmpire.

Conceived as a method for Java to 'call the shots,' the jUmpire is an over-
arching method of mediating communication between Java and MS code bases.
To get a better understanding of where we are going, march to sesma.com



ALL OF THE PRODUCTS LISTED HERE ARE PART OF THE
SESMA INTERACTION PLATFORM (SIP) AND/OR
APPLICATION DISTRIBUTION PLATFORM (ADP)

SESMA

Client/Server Is Not the Only Fruit

by Dan North

The vast majority of Java enterprise applications are architected along the lines of Sun's original PetStore showcase application. Rather than seeing this as a market stall displaying all the J2EE goods on offer, developers took it as a blueprint for enterprise applications. It certainly created a lot of work for Java application developers (as well as systems administrators and systems integrators), but whether most of the applications needed most of the possible layers – including bean-managed persistence and the ubiquitous DAO – is open for question at the very least. This article briefly describes some of the testability shortcomings of the traditional J2EE model and presents some alternatives.

The J2EE Model

With J2EE we are tied to a single logical server in a client/server architecture, and any component such as a servlet or EJB has to be deployed to the server (potentially a time-consuming process) before we know whether it works. This makes it difficult to use test-driven techniques to develop these components. To make a small change to such a component requires a code change, a recompile, and a redeploy. On many commercial application servers this sequence of events takes too long to be repeated as frequently as TDD requires. The developer is encouraged to deploy bigger changes less often, and therefore has a higher risk of introducing errors.

A few patterns alleviate some of this, most notably the Front Controller/Dispatcher pattern. This has a single component (a servlet in the Web container or a stateless session bean in the EJB tier) that dispatches requests to a *POJO* (Plain Old Java Object). A POJO is much better suited for unit testing – we can simply instantiate it in a known state and then exercise its behavior. From a TDD perspective, this is what makes a framework like WebWork more “agile” than Struts; for instance, the actual components are testable outside of a container.

Service-Oriented Architectures

J2EE encourages us to model a system in terms of components interacting with one another. Instead we can model in terms of services that an application provides or

“things that it does.” These services can exist on different machines or in different virtual machines on the same host. As long as we have a means of locating a service, we can use it. In terms of testability, this makes life a lot easier for us. A service is simply a class that implements an interface, so we can test its behavior in isolation or in conjunction with other services just by instantiating a service object and calling methods on it.

Service-oriented mechanisms such as SOAP, XML-RPC, or JINI are cleanly separated from the actual services they advertise. Compare this to a session bean, which is tightly coupled to its container: it requires a number of container-specific classes just to function (such as a `javax.ejb.`

`SessionContext`), and has a whole complex life cycle associated with it. Both of these make testing a session bean a lot more complex – and add greatly to the time it takes to make a change and test it.

Event-Driven Architectures

So far we have focused on synchronous behavior. The concept of modeling a system in terms of messages or events being produced and consumed asynchronously from queues is very powerful and yet still largely overlooked. Event-driven systems are renowned for their resilience, but they also lend themselves very nicely to a test-driven approach. As with services, an event consumer or listener is typically a Java class that implements an interface. As long as the consumer class is decoupled from the delivery mechanism,

we can easily test its behavior. We can model synchronous behavior asynchronously to gain both scalability and testability. A Staged, Event-Driven Architecture (SEDA) simulates synchronous processing by means of a series of stages, each represented by a queue. Thus each stage can be unit-tested and developed separately.

Summary

When you're writing an enterprise application in Java, you shouldn't feel constrained to the classic J2EE model. Investigate some of the alternatives; ask lots of questions; and evaluate the costs, benefits, and risks of using a different approach. But most important, as a developer try to find an architecture that enables you to write testable code. Remember, PetStore is a showcase of technologies, not a blueprint. ☼



Dan North has been writing software for 12 years, and is a programmer and coach for ThoughtWorks (www.thoughtworks.com), a software development consultancy, where he encourages people to write tests.

dan.north@thoughtworks.com



©2003 Wily Technology, Inc. The Wily logo is a trademark of Wily Technology, Inc. Java is a trademark of Sun Microsystems in the U.S. and other countries.

Two years without a vacation.
The application's up. It's down.
It's up. It's down.

I'm to blame. Steve's to blame.
Someone's always to blame.

Not any more.

Get Wily.™

Enterprise Java
Application Management
1 888 GET WILLY
www.wilytech.com



Clustered Timers

FOR ROBUST, SCALABLE SYSTEMS

by Clark Richey, Jr.

Often, when someone asks how we are going to scale the Web application we're about to develop, we look at them, smile, and say, "Not a problem – we'll just cluster the application servers." Clustering our application across multiple servers provides us with the ability to handle large volumes of traffic and to scale systems by adding additional servers to the cluster.

In addition to providing scalability, application clusters make the system more robust by allowing for automatic system failover when a server fails. This way when one server goes down the application continues to run, albeit with slightly decreased performance. While it is true that the current generation of application servers makes it relatively pain-free to create a cluster, there are still several significant, if often overlooked, design issues that must be taken into account now that the system is clustered.

When we run our Web application in a cluster, we have the exact same software running on each machine in the cluster. While this eliminates a host of configuration management difficulties, it does create other problems. While we don't have to write different code for every possible machine in the cluster, there are times when this simplicity actually makes things more complex; the running of scheduled tasks is typically one of these areas. Scheduled tasks are used to execute procedures that need to run at certain fixed times or at fixed intervals. Typical examples of scheduled tasks within a Web application are report-generation tasks and tasks that send data to external systems that are only available within a certain time frame.

To understand why clustering affects how we design our application to handle scheduled tasks, let's consider a generic e-commerce Web application. To allow management to analyze sales trends, profits, inventory, etc., the system has been set up to periodically compile a set of reports and e-mail them to management. Clearly, management doesn't want to receive multiple e-mails containing the same reports, yet this is what we will get if we simply write a scheduled task and then cluster our system. When the appointed time to run the report comes up, all machines in the cluster will generate the same report

and send it to management. This can be seen visually in Figure 1.

Perhaps the most straightforward way to solve this problem is to package the code that runs the scheduled tasks into a separate JAR file within the EAR file that contains the WAR file for the Web application. This EAR file is deployed to all the servers in the cluster; however, the JAR containing the scheduled tasks is configured to run on only one of the servers. This solves the problem by preventing the scheduled tasks from ever running on multiple machines. However, there is a significant downside to these solutions. First, you have now created additional configuration management problems. You need to carefully track which servers are set up to run the scheduled tasks and the exact deployment procedures that were used so that when additional servers are added to the cluster, the application is properly deployed on those servers.

The second problem is that you have effectively taken the scheduled tasks out of the cluster. Now, if the machine that is set up with the scheduled tasks fails, or its connection to the network fails, there is no backup or failover system. The tasks won't run. The remainder of this article investigates solutions to this problem that allow the scheduled tasks to remain part of the cluster and don't involve additional configuration management.

To stop every system in the cluster from performing the same scheduled task, report generation in this case, we have to utilize something outside of the application server cluster to track the state of our scheduled task. A perfect candidate for maintaining the state of our scheduled tasks



Clark Richey, Jr., works for Aquilent, Inc., as a senior technical consultant on a variety of government projects. Clark is currently interested in examining distributed computing and alternatives to browser-based applications.

crichey@speakeasy.net

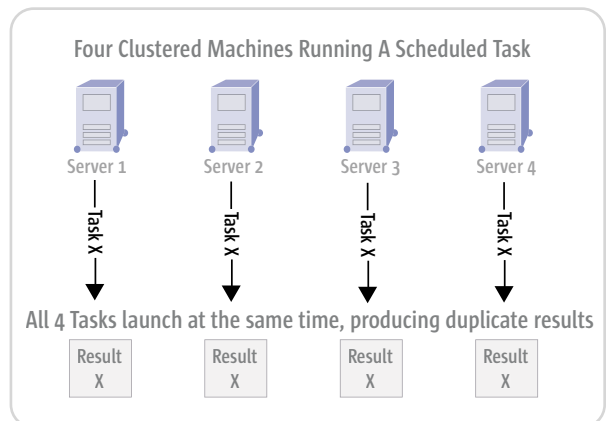


Figure 1 Four clustered machines running a scheduled task

is a shared database, and since nearly all applications already have access to a shared database, this is the resource we will use to solve this problem in our example (see Figure 2). It's worth mentioning that while a shared database is an ideal resource for solving this problem, it's not the only option. The solution presented here could be adapted to use flat files or some other shared resource external to the cluster.

Our external resource, the database in this case, will act as a mediator between competing machines in the cluster. We will create a table in the database that tracks scheduled tasks and their status. When a machine in the cluster wants to run one of the scheduled tasks, it first checks the status of that task in the database to see if some other machine is already running that task. If no other machine is running the task, the status of the task will be updated and that machine will run the task.

Another way of thinking about this solution is to think in terms of a concurrent method running on a single machine. If we see the scheduled task in these terms, it becomes clear that the best way to keep multiple threads from running the task at the same time is to use some sort of semaphore. Again, if this was a single method on one machine, we could easily do this by creating a synchronized block around the code that we wanted to protect. When a thread first attempts to enter the synchronized block, it has to attempt to get the lock. If it fails to get the lock, it can't run. In our distributed system, we are using the database as the lock.

We will call our database table "Tasks" and it will have three columns. The first column will be the name of the task, the second the status of the task, and the third the date and time that the task last changed status. The generic SQL script to produce this table is shown below.

```
CREATE TABLE `Tasks` (
  `TaskName` varchar(50) NOT NULL,
  `Status` varchar(25) NOT NULL,
  `StatusTime` datetime,
  PRIMARY KEY (`TaskName`)
);
```

Now that we have created our database table to serve as our mediator, we can create the class that accesses this table in order to determine if a particular instance of a Task can execute. We'll call this class TaskMonitor. (The source code for this article can be downloaded from www.sys-con.com/java/sourcecc.cfm.) The class exposes two methods to the public, public static boolean acquireLock(String taskName) and public static void releaseLock(String taskName). Before a Task runs, it will need to call the acquireLock method of the TaskMonitor. If this method returns True, it's safe for the Task to run. If it returns False, then it's not safe for the Task to run as some other instance of this Task in the cluster is already executing. The key to understanding the TaskMonitor class is to understand the ACQUIRE_LOCK SQL query on lines 5-7.

What needs to be done is to determine if the Task in question, as identified by the field TaskName, is currently Idle, and if so, change its Status to Active. The crucial aspect of this is that it needs to happen atomically, that is, it must all happen as one single step. That's why we use a single update statement instead of writing both a select statement to see if the Task is currently Idle and an update to change its Status. In the case where we use the select statement first, it would be possible for the same select

statement to be run by the other machines in the cluster before the update is executed. This would result in multiple Tasks running since they would all see the Idle state. By performing the entire process in an update statement, we take advantage of the automatic exclusive row locking that takes place in the database whenever an update statement is executed.

Now that we understand how the ACQUIRE_LOCK query works, the rest of the acquireLock method of the TaskMonitor is easy to follow. On line 23 the query is executed and the results are examined. The executeUpdate method returns the number of rows that were affected by the query. When the ACQUIRE_LOCK query successfully changes the Task from Idle to Active (as will be the case when this particular query is the first one in the cluster to run), one row will have been affected and the lockAcquired flag will be set to true. Otherwise, no rows will be affected and the lockAcquired flag will remain false.

The releaseLock method of TaskMonitor is meant to be called when a Task has finished executing. This method simply changes the status of the Task back to Idle. Both the releaseLock and the acquireLock methods also update the StatusTime field with the current date and time for record-keeping purposes.

One final note on the TaskMonitor class: the getConnection method shown in lines 75-85 should be upgraded before placing this class into production. As written, the method creates a connection to an instance of a MySQL database. A better practice in production would be to retrieve a connection from an existing connection pool.

Together the Tasks database table and the TaskMonitor class provide a framework for ensuring that only one instance

Build Incredible Interactive Diagrams
Feature Packed New JGo Release!

JGo, written entirely in Java, makes it easy for your Java applications to create interactive diagrams of all kinds. Create network editors, schematics, process flow diagrams, visual languages, organization charts, family trees, etc.

The JGo class library provides containers, connectors, links, orthogonal routing with automatic node avoidance and enhanced link options, drag-and-drop, scaling, in-place text editing, tooltips, printing, SVG/XML, nested graphs, lots of new sample nodes and applications, a high performance Auto Layout Option, improved IDE integration, and more...

- **FREE Evaluation Kit**
- **No Runtime Fees**
- **Full Source Code**
- **Easy to Learn & Use**
- **Excellent Support**

New Version 5.0!

www.nwoods.com/go/
800-434-9820
 Ask us about SWT!

Northwoods SOFTWARE CORPORATION

of a given Task is running at a particular time, no matter how many instances of the application are running within the clustered system. At this point we're ready to create our report generating Task.

Because we're concerned with managing Tasks in a clustered environment, and not with creating reports or using the `javax.mail` APIs, we'll create a simple Task, called `ReportTask`, to illustrate the concept. Because we want this Task to execute automatically on a schedule, we need to extend `java.util.TimerTask`. `TimerTask` is an abstract class that has one method that we have to implement, `public void run()`. This is the method where all the Task's work is done. For our simple example, `ReportTask`, we'll output some text to show that the Task is running. The code for this class is shown below.

```
1) import java.util.TimerTask;
2) public class ReportTask extends TimerTask {
3)     public void run() {
4)         if(TaskMonitor.acquireLock("ReportTask") == false)
5)             return;
6)         System.out.println("Creating report to be emailed...");
7)         TaskMonitor.releaseLock("ReportTask");
8)     }
9) }
```

The key thing to note here is that before the `ReportTask` actually performs its work, printing some text in this case, it first attempts to acquire the lock for this Task by making the call to `acquireLock` on line 4. If it fails to acquire the lock, it simply returns without performing its work. However, if it does successfully acquire the lock, then it's free to perform its work and it goes ahead and prints out its message on line 6. Once the Task is complete, it's vital that the lock be released. This is accomplished by calling `releaseLock` on line 7. If the lock is never released, this Task will never run again on any machine in the cluster. Ensuring that the lock is properly released is clearly not an issue with this simple example; however, in more

complex tasks it can be tricky. Consider a Task where several different error conditions could cause the Task to terminate before running to completion. There are now potentially several places where the lock will have to be released.

At this point, you've probably noticed a serious problem with our Task. We never populated the `Tasks` table with any tasks. As things stand, our `ReportTask` will never be able to acquire a lock and will never run, and this step needs to take place for every Task that's going to be managed in this way. To rectify this situation we need to insert the `ReportTask` information into the `Tasks` table using the following SQL script:

```
insert into Tasks values ('ReportTask', 'Idle', null);
```

We've nearly finished setting up our system for managing clustered tasks. So far we've created an external resource and a `TimerTask` called `ReportTask` that will run in our cluster. All that remains to be done is to create a `Timer` for running our `ReportTask`. Because we want to start the `Timer` for our task as soon as the application starts, we'll create a servlet called `StartupServlet` that does the work of creating our `Timer`. We will ensure that `StartupServlet` is loaded immediately by adding the following lines to `web.xml`:

```
<servlet>
  <servlet-name>StartupServlet</servlet-name>
  <display-name>StartupServlet</display-name>
  <description>Used to create the Timers</description>
  <servlet-class>StartupServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

As our simple `StartupServlet` is not designed to handle requests, it doesn't need to override any method other than `init()`. When we create the `Timer` for running our `ReportTask`, it's important that we use one of the overloaded constructors to create the `Timer` as a daemon thread. If we don't specify that the `Timer` should be a daemon thread and use the default `no` argument constructor, the `Timer` will not be a daemon thread. By making it a daemon thread, we ensure that the `Timer` will continue to run for as long as our Web application runs and that it will terminate when the application terminates. We don't want to try to generate reports if the application has been stopped for some reason.

After calculating how many milliseconds are in a day (we want our `ReportTask` to run once a day), we schedule the `ReportTask` to run daily, starting now. On line 12 we place the `Timer` that we created in the `ServletContext`. While this is not strictly necessary to keep the `ReportTask` running, by keeping a reference to the `Timer` available we are able to check easily on the status of the `ReportTimer` or cancel it entirely should the need arise.

With the `StartupServlet` in place, we now have a very basic but workable system for running scheduled Tasks in a clustered environment, without having to worry about the same task running on all of the machines in the cluster simultaneously. It's important to note that if this scheme is used as presented and the tasks being executed complete in a very short period of time, you could still see duplicate executions of the same task if the clocks on all of the machines are not in synch with each other. While it is possible to extend this approach to address this problem, it's outside the scope of this article. With a little bit of effort, this system can also be extended to allow for such things as programmatic modification of the running tasks, robust error handling, and recovery of frozen tasks. ☺

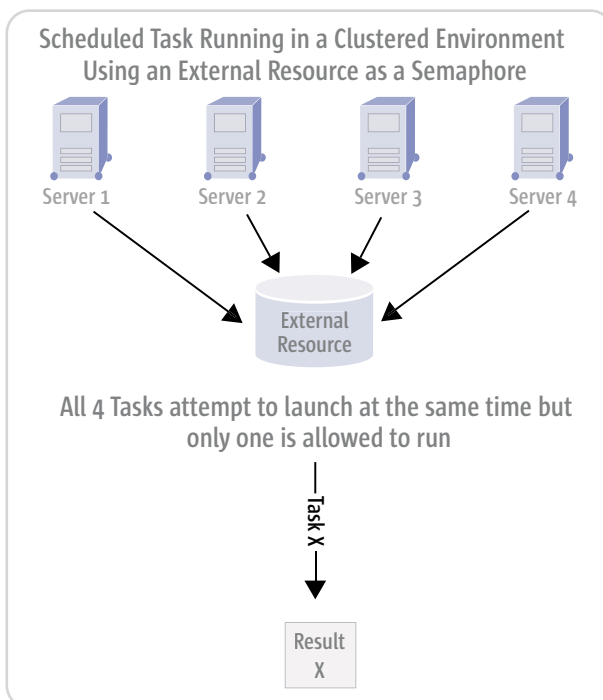


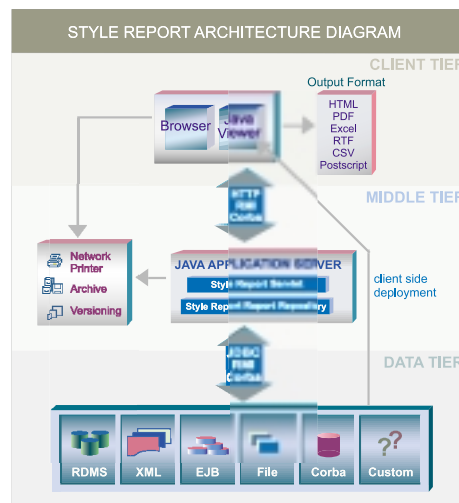
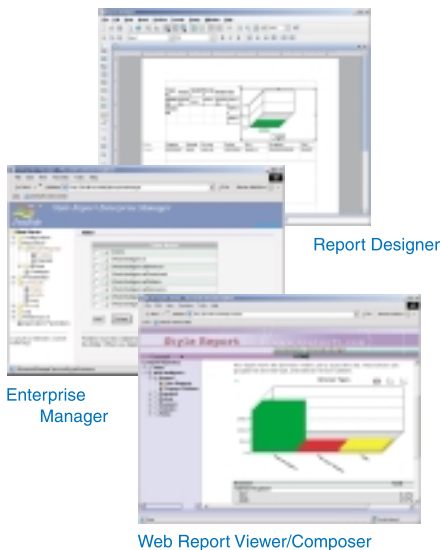
Figure 2 Scheduled task running in a clustered environment using an external source as a semaphore

Java & Web Reporting: with Java flexibility without Java complexity



Style Report: Information Vantage Point

Style Report blends the best of Java and Web to provide an award winning Web enterprise reporting solution. Based 100% on open standards such as XML, JSP, Java and Web services, Style Report allows you fully leverage existing IT infrastructure and common technical skills. One Report, Multiple Channel delivery and zero client ad-hoc reporting gives end users the freedom and frees IT from time consuming repetitive work. Designed for integration, Style Report is deployed and managed as a standard Web application or Corba/RMI service requiring not special skills. Download a free evaluation copy now!



Style Report Features

Flexible Visual Designer with JavaScript or VB Script	<input checked="" type="checkbox"/>
PDF, Excel, Word, CSV and other Export Format	<input checked="" type="checkbox"/>
Database/XML and other Datasources	<input checked="" type="checkbox"/>
Zero Client Browser Viewing	<input checked="" type="checkbox"/>
Built-in Drilldown/Parameters	<input checked="" type="checkbox"/>
Zero Client Ad-hoc Reports	<input checked="" type="checkbox"/>
Virtual Private Security Model	<input checked="" type="checkbox"/>
J2EE Drop-in Deployment	<input checked="" type="checkbox"/>
Advanced Scheduling	<input checked="" type="checkbox"/>
Report Versioning/Archive	<input checked="" type="checkbox"/>
Demand Paging/Report Cache	<input checked="" type="checkbox"/>
Portal/JSP integration	<input checked="" type="checkbox"/>

J2SE 1.5

Tiger debuts with beta

by Arulazi Dhesiaseelan

The beta release of the Java 2 Platform, Standard Edition (J2SE) 1.5, started gaining momentum in the developer community due to its potential improvements to the language and its convincing feature set. J2SE 1.5, code named “Tiger,” is being developed under the Java Community Process (JCP). The umbrella Java Specification Request (JSR) for this release is JSR 176. This article outlines the major features that are shipped with J2SE 1.5.0 Beta 1, which was released in February 2004. The final RI and TCK are targeted to be released in the summer of 2004.

Component JSRs in Tiger

JSR 176 defines the release contents of Tiger. There are at least 15 component JSRs that are developed under the JCP program and targeted toward the Tiger release. Table 1 details the JSRs along with their status as of this writing.

Detailed information about these JSRs is available at www.jcp.org/en/jsr/detail?id=JSR_NUMBER; simply replace the JSR_NUMBER with the actual JSR ID you are interested in. The status information mentioned in Table 1 was obtained from the individual JSR page in the JCP Web site. Some of this status information may not be current.

Generics

What Are Generics?

Generics are also called as parameterized types (often referred to as type polymorphism). To be more precise, Generics are similar to the powerful C++ templates but don't have the potential drawbacks faced by templates. Generics make type parameters explicit and type casts implicit, making the use of libraries safer and more flexible (e.g., the collections library).

The generics support in Java is the most frequently requested language feature from Java developers. This feature has been postponed for quite some time as it requires changes to the Java language specification, and also updates to the Java Virtual Machine (JVM) specification. Finally, this feature stands as the top-most priority for the Tiger release.

Generics Types

There are two forms of generics types.

- Parameterized types
- Type variables

A parameterized type consists of a class or interface type *C* and a parameter section $\langle T_1, \dots, T_n \rangle$. *C* must be the name of a parameterized class or interface; the types in the parameter list $\langle T_1, \dots, T_n \rangle$ must match the number of declared parameters of *C*; and each actual parameter must be a subtype of the formal parameter's bound types.

As stated in section 2.1 of “Adding Generics to the Java Programming Language: Participant Draft Specification” dated April 27, 2001: “a type variable is an unqualified identifier. Type variables are introduced by parameterized class and interface declarations and polymorphic method declarations.”

The generics syntax allows developers to enforce parameters, extend a class, or implement an interface. This helps restrict the type of parameters for a generic class/interface/ method and they are referred to as bound types. Listing 1 details the usage of generics in parameterized types, classes, interfaces, and methods.

Generics Advantage

The following code snippet shows the usage of generics in collection libraries. When using generics, you need to define the type the collection can hold. This helps the collection maintain homogeneous objects. If an undefined type is added to this collection, you'll be notified of errors at compile time. This helps us identify the cause of the problem during development time.

On the contrary, the traditional way of using heterogeneous objects in a collection may result in runtime errors. Sometimes these errors are uncovered during deployment, which is very crucial to the product quality.

With Generics

```
Map<String, Integer> map = new HashMap<String, Integer>();
map.put("one", new Integer(1));
Integer integer = map.values().iterator().next();
```

Without Generics

```
Map map = new HashMap();
map.put("one", new Integer(1));
Integer integer = (Integer)map.values().iterator().next();
```

Thus the generics approach is more readable and powerful than the traditional approach.

Collection Filtering

Filtering a collection is prone to ClassCastException when the collection elements are heterogeneous, and may fail at runtime due to invalid casts. This behavior is overcome by the use of generics in collection filtering, which helps developers detect the errors at compile time. Listing 2 shows the current collection filtering and filtering collection with generics. (Listings 2-6 can be downloaded from www.sys-con.com/java/sourcecc.cfm.)

Getting Started with Tiger

Download the J2SE 1.5.0 Beta 1 release from <http://java.sun.com/j2se/1.5.0/download.jsp>.

By default, the installer places the JDK directory under “C:\Program Files\Java\j2sdk1.5.0”. The Java Language Specification (JLS) implemented by the javac compiler in the Tiger beta release is version 1.4. To play with the new language features, its required to pass the argument -source 1.5 to the javac command.

IDEs that Support Generics and Other New Language Features

IDEs are available for writing generics-enabled code today, in addition to other new language features:

- *Omnimac CodeGuide 6.1*: www.omnicore.com/codeguide.htm
- *IntelliJ IDEA 4.0*: www.jetbrains.com/idea/
- *Eclipse Java IDE*: www.eclipse.org/jdt/index.html



Arulazi Dhesiaseelan

is a senior software engineer for Hewlett Packard Company, India.

He has a master's degree in computer applications from PSG College of Technology in India.

He has about four years of industry experience and was also involved in the UDDI4j project hosted at <http://uddi4j.org>. He has been working on Web services-related technologies such as WSDL, UDDI, and SOAP. Currently,

Arulazi is involved in developing a standards-based mobile infrastructure for mobile network operators.

aruld@acm.org

Typesafe Enumerations

What Is Enum?

Enum is a special kind of class declaration. A typesafe enum facility defines a fixed set of constant legal values for types. This feature is borrowed from the C language and has been incorporated into the Java language in the Tiger release.

Advantages of Enum

The proposed enum facility has more advantages when compared to the traditional int enum approach in Java.

The advantages are:

- Enum provides compile time type safety.
- Enum provides proper name spacing for its types.
- Performance is comparable to int constants.
- Typesafe constants do not require a compilation when clients modify constants.
- Printed values are informative.
- Enum can be used in collections as they are objects.
- Enum is a special kind of class declaration, hence you can add arbitrary fields and methods.
- Enum can be made to implement arbitrary interfaces.

Traditional Enum Pattern

The commonly used pattern for enumerated types in Java suffers from many drawbacks as shown in the following code. In this approach, whenever you modify the client code, you need to recompile.

```
public class Continent {
    public static final int CONTINENT_AFRICA    = 0;
    public static final int CONTINENT_ASIA     = 1;
    public static final int CONTINENT_EUROPE   = 2;
    public static final int CONTINENT_NORTH_AMERICA = 3;
    public static final int CONTINENT_OCEANIA  = 4;
    public static final int CONTINENT_SOUTH_AMERICA = 5;
}
```

Typesafe Enum Pattern

Java provides an alternative method to overcome the drawbacks of the traditional approach, called the typesafe enum pattern. This pattern defines a class that represents a single element of an enumerated type that doesn't provide any public constructors to it. Provide static final fields for each constant in the enumerated type. Listing 3 shows this pattern approach. This pattern provides compile time type safety by allowing only the six continents defined to be passed to a method that defines a parameter of type Continent.

Proposed Enum Facility

The proposed facility is simple and readable. The construct can be used with switch statements. Listing 4 shows the usage of the proposed enum facility for the Java language.

Autoboxing

What Is Autoboxing?

Developers are burdened when they convert primitive types to wrapper (reference) types (for example, converting int to Integer type). Adding primitive types to a collection is not possible unless they are converted to the corresponding reference type. To overcome this drawback of the cur-

rent type system, the need for the automatic conversion of primitive type data to their corresponding wrapper type data was proposed. This conversion mechanism is known as autoboxing.

Listing 5 illustrates the code that achieves autoboxing in a collection. In this code sample, primitive type int is added to a collection that holds an Integer reference. The compiler takes care of type conversion for you.

There are several other forms of conversion that are supported by the compiler. Refer to JSR 201 for more information (<http://jcp.org/aboutJava/communityprocess/jsr/tiger/autoboxing.html>).

Enhanced For Statement

What's Wrong with the Current For Statement?

The current for statement is efficient and powerful in all aspects, but it's not optimized when it iterates over a collection, as the iterators are used only to get elements out of a collection and serve no other purpose. Generics helped make this situation better by adding the type safety to the collection. This resulted in proposing "enhanced for" statements with generics additions to them. With this new feature, the compiler takes care of the iterator for you. Listing 6 details the usage of "enhanced for" statements.

Using an "Enhanced For" Statement in an Int Array

The "enhanced for loop" has been specifically designed for iteration over collections and arrays. Its usage in an integer array is shown below:

```
public class TestIntegerArrayUsingEnhancedForLoop {
    public static void main(String args[]) {
        int [] a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        int sum = 0;
        for (int e : a)
            sum += e;
        System.out.println("Array Sum : " + sum);
    }
}
```

Static Import

Rationale for Static Import

In Java, when using any mathematical functions from the "java.lang.Math" package or when using named constants, you have to prefix the method name or field name with the class name. This way of coding looks more verbose. To make it more convenient for developers, the concept of static import was proposed. Using this, you can allow the import of static methods and fields similar to the import of classes and interfaces.

```
import static java.lang.StrictMath.*; // import static
import static java.lang.System.*; // import static

public class StaticImport {
    static void testStaticImport() {
        out.println("The square root of PI is : " + sqrt(PI));
    }
    public static void main(String args[]) {
        testStaticImport();
    }
}
```

In the above code snippet, the usage of static import is illustrated. The traditional approach is discussed below.

When accessing a static method, you need to prefix with the class name, for example:

```
Math.sqrt(Math.PI);
StrictMath.IEEEremainder(4.0, 2.0);
System.out.println("hello");
```

When accessing named constants, you need to prefix with the class name. For example:

```
class Suit {
    public static final int CLUBS = 0;
    public static final int DIAMONDS = 1;
    public static final int HEARTS = 3;
    public static final int SPADES = 4;
}

Suit.CLUBS
Suit.DIAMONDS
Suit.HEARTS
Suit.SPADES
```

This approach makes code more verbose. The static import helps you overcome this approach. This is a simple yet convincing feature for developers.

Conclusion

The features discussed in this article are only a subset of the features that are shipped with the Tiger beta release. These features are developer savvy and powerful additions to the Java programming language. The listings in this article are tested with J2SE 1.5.0 Beta 1 release. JSR 176 is currently available for public review at www.jcp.org/en/jsr/detail?id=176.

References

- *Java 2 Platform, Standard Edition 1.5.0 Beta 1 release*: <http://java.sun.com/j2se/1.5.0/download.jsp>
- *J2SE v 1.5.0 Beta 1 Documentation*: <http://java.sun.com/j2se/1.5.0/docs/index.html>
- *JSR 176, J2SE 1.5 (Tiger) Release Contents*: www.jcp.org/en/jsr/detail?id=176
- *JSR 14, Add Generic Types to the Java Programming Language*: www.jcp.org/en/jsr/detail?id=014
- *JSR 201, Extending the Java Programming Language with Enumerations, Autoboxing, Enhanced For Loops, and Static Import*: www.jcp.org/en/jsr/detail?id=201

Listing 1: Generics usage in parameterized types, classes, interfaces, and methods

Parameterized Type Example

```
List<Integer> integerList = new LinkedList<Integer>();
Map<String, Integer> integerMap = new
HashMap<String, Integer>();
Stack<Integer> integerStack = new Stack<Integer>();
```

Generic Class Example

```
public class C<T1, T2> {
    private T1 type1;
    private T2 type2;
    public C(T1 type1, T2 type2) {
        this.type1 = type1;
        this.type2 = type2;
    }
    public T1 getType1() {
        return this.type1;
    }
    public T2 getType2() {
        return this.type2;
    }
    public static void main(String args[]) {
        C<String, Integer> cSI = new C<String, Integer>("one", 1);
        String type1SI = cSI.getType1();
        int type2SI = cSI.getType2().intValue();

        C<Integer, Boolean> cIB = new C<Integer, Boolean>(1,
true);
        int type1IS = cIB.getType1().intValue();
        boolean type2IS = cIB.getType2().booleanValue();
    }
}
```

Generic Interface Example

```
public interface I<T> {
    public T getConnectionPool();
    public void releaseConnectionPool(T connectionPool);
}
```

Generic Method Example

```
public class M {
    public static <T extends Comparable> T minimum(T a, T b) {
        if (a.compareTo(b) <= 0) return a;
        else return b;
    }
    public static void main(String[] args) {
        Integer b1 = new Integer(2);
        Integer b2 = new Integer(5);
        Integer min = minimum(b1, b2);
        System.out.println("Minimum of (2,5) : " +
min.intValue());
    }
}
```

JSR ID	Description	Current Status
003	Java Management Extensions (JMX) Specification	Final Release 3
013	Decimal Arithmetic Enhancement	Proposed Final Draft 2
014	Add Generic Types to the Java Programming Language	Public Review
028	Java SASL Specification	Final Release 2
114	JDBC Rowset Implementations	Proposed Final Draft
133	Java Memory Model and Thread Specification Revision	Public Review
160	Java Management Extensions (JMX) Remote API 1.0	Final Release
163	Java Platform Profiling Architecture	Public Review
166	Concurrency Utilities	Public Review
174	Monitoring and Management Specification for the Java Virtual Machine	Public Review
175	A Metadata Facility for the Java Programming Language	Public Review
200	Network Transfer Format for Java Archives	Public Review
201	Extending the Java Programming Language with Enumerations, Autoboxing, Enhanced For Loops and Static Import	Public Review
204	Unicode Supplementary Character Support	Community Draft Ballot
206	Java API for XML Processing (JAXP) 1.3	Public Review

Table 1 Tiger Component JSRs



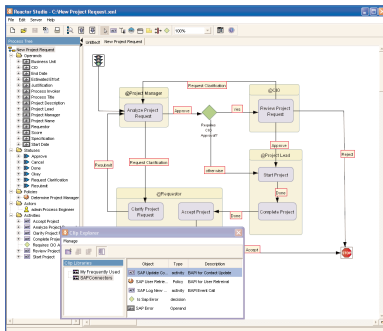
Welcome To Workflow Nirvana

The Workflow Functionality You Want The Ease of Use Your Business Users Demand

Combining the control, extensibility and platform neutrality developers want — with the ease of use business users need, the Reactor 5.5 workflow engine has at long last arrived to help everyone co-exist in blissful peace and harmony.

Available as royalty-free source code, Reactor 5.5 provides:

- A single, scalable process platform for defining and linking enterprise-wide workflows
- Extensible libraries of re-useable process components
- A user-friendly tool for point-and-click development of custom workflow solutions.



Process design tool, Reactor Studio, enables quick development and deployment of workflows.

So if you're seeking the path to true enlightenment (and killer gains in productivity) — call us at (818) 440-1234, or download a **FREE** evaluation copy at www.oakgrovesystems.com/jdj.

We'd love to help you...

Declare Your Workflow Independence!™



JDJ Industry Profile:

Oracle's
Thomas Kurian

SENIOR VICE PRESIDENT, DEVELOPMENT
ORACLE APPLICATION SERVER

Pursuing the
Dream

DATABASES, MIDDLEWARE, THE GRID, AND MORE...

by Jeremy Geelan

Two and a half years ago, when he stepped in to help Oracle get back into the application server business, Thomas Kurian had quite a tough row to hoe. Now he has 19,000 customers. **JDJ** explores with Kurian his 30-month journey, and asks too about other aspects of emerging technology seen from the Oracle perspective.

In **JDJ** last month, Sun's Jonathan Schwartz claimed the "death" of middleware, so my first question to Thomas Kurian is inevitable, given his role as a champion of middleware for Oracle. What does he make of Schwartz's claim?

"Those who say such things are being facetious," replies Kurian, bluntly. "A company that says that there is no role for middleware means it in the sense that it feels its own operating systems incorporate middleware facilities. Such a company is bundling proprietary middleware into the OS as a vehicle to try and lock customers in."

Kurian compares the current situation vis-a-vis middleware with relational databases in the early days.

"Early on, relational databases were primarily just query engines, and the information management infrastructure was very rudimentary," he explains. "Over a 20-year period, relational databases have incorporated a number of specialized features and functionalities – clustering capabilities, data warehousing capabilities, security for the data, and so on – and today you might argue there is certainly a lot of change still in database computing, but people see it as a much more mature infrastructure for information management."

"Now with first-generation middleware," Kurian continues, "the killer application was to deploy processing logic into a J2EE server and scale up the application so lots of clients can access it."

Now he believes that what we're seeing is an "evolution" of middleware.

"During the early phase of such an evolution obviously the rate of change is higher, but over a period of time these technologies mature and at some time in the future – in two, three, five years, I don't want to make a prediction – there will be a much more mature application infrastructure, and while there will continue to be change it will be at a lesser rate than today."

Does Oracle see greater opportunities in this mature phase, I wondered.

"Part of the process of maturation," replies Kurian, "is that customers want to eliminate all the piecemeal componentry that they had been using and shift to a more mature platform that offers a broader set of services."

"When we entered the market two and a half years ago with our integrated application server we were the first to integrate all these pieces into one app server. That to us is the biggest opportunity from a business point of view. The customers that we talk to validate this again and again – telling us how they save cost and time, either from a development perspective because they can use a single development tool

and framework to do a lot more things, or from a deployment and management perspective because they have a much simpler and easier-to-manage infrastructure. It saves them a lot of cost and gives them a lot of value and that's why we continue to invest so heavily in making this vision a reality."

Can the Oracle vision be unpacked in ways that differentiate it from the other giants, like IBM, BEA, and Sun? Kurian was very precise when talking about the various distinguishers.

"When you look at Oracle Application Server versus IBM," he says, "our middleware has always been built as a single cohesive development tool with a single cohesive runtime platform, not a brand like WebSphere with many different pieces, but a single integrated product. It is this degree of cohesion that is the big difference compared to IBM."

"Compared to BEA," he goes on, "the biggest difference is the breadth of our capability. We are broader in two dimensions. First, in terms of deployment and management: we have much more sophisticated capabilities in the way we can manage the entire application server and all of its services – monitoring, life-cycle deployment, software provisioning, identity and security management – we make all of that much easier. Second, we have more capabilities than WebLogic around Web services, wireless access, business intelligence, and so on."

And what of the last of the Big Three rivals, Sun?

"Compared to Sun, the first big difference is that our application server is actually used by lots of customers in a heterogeneous environment, not just on Solaris but on lots of different OSes – Solaris, HP, Windows, name your favorite. They access Oracle and non-Oracle databases; they use it to access mainframe systems; they use it to access applications like SAP. We have 19,000 customers using it in a highly heterogeneous, multivendor environment while Sun typically pitches its product at Solaris or Sun shops."

In the pecking order, Kurian says, Oracle feels it's in second place with the number of customers – with 19,000 plus, whereas BEA has only about 15,000.

"In terms of number of new customers," he adds, "every quarter we add 1,500–1,700 new customers, pretty consistently. So we feel very comfortable that in terms of real deployment we're number two in the market."

There's some growth in the overall market, he says; application servers didn't decline, but they were pretty flat during the recent tough market conditions. The downturn had an impact on smaller companies, naturally.

Java's Mis-step

"If you look at the last 5 or 6 years," Kurian says, "Java initially took a detour defined by browser-centric Java. I think that was a mis-step. Java's real value is on the server side. I think J2EE helped to establish that by providing, for the first time in the industry, a portable programming model on the server side, with business logic and applications."

"The paradox is that as the new versions of J2EE standards come out – 1.3, 1.4, 1.5 – J2EE is encompassing more and more capability to allow the Java programmer to do more things. Things that, for example, in 1.5, EJB 2.1, the JMS enhancements, give the developer more capabilities to do enterprise application integration a lot better. Enterprise portals with the new JSR standards are allowing the Java developer to solve more business problems than their companies actually have!"

Is the paradox that even as J2EE incorporates a number of these capabilities, it needs to be kept accessible to those people who want to do simple things simply?

“Yes,” says Kurian, “there’s a risk that J2EE gets so complicated and so cumbersome that people who want to build very simple applications find it challenging.

“That’s where I believe personally that some of the new efforts over, say, JavaServer Faces help,” he adds.

“Compared to .NET, which spends a lot of time talking about how to build presentation-centric applications, with WinForms and .NET forms and things like that, J2EE – other than JSP – has really not had a great model for presentation-oriented kinds of applications. These are what a lot of people want to get going with; they want to build simple, forms-style applications.”

Oracle participates very actively in the J2EE process, says Kurian. He has about 150 people in his organization who participated in about 139 JSRs in the last year alone.

“We feel that the Java Community Process is the right model for Java. The community has to find a way to leverage the best efforts of people and make this problem of complexity self-correcting.”

Of course, the JCP is not without its drawbacks and limitations. “Some vendors play games with the community process,” Kurian concedes, “trying to take control of it and take it in specific directions in an attempt to favor their product, as opposed to making sure that Java retains both the nascent appeal that it has for people as the only solution that allows

corporate-wide business processes, for business process automation.”

But there have been limitations to the way that Web services has played out, reckons Kurian.

“Where things have fallen down is twofold,” he observes. “First, Web services was overhyped by Microsoft and IBM as a technology that you could use to do anything you needed, and there’s always some disillusionment after such hype.

“Second, there’s been a real slow process in terms of the implementation and acceptance of Web services standards, particularly in areas like reliable messaging, security, coordination and activation, process management, orchestration, as well as things like policy.”

In Kurian’s view, these are the two main elements that have hindered their adoption today within enterprises.

Grid Computing

Is there a danger of similar hype with grid computing, the coordinated use of servers and storage acting as one large computer, I ask.

“When Oracle talks about the grid,” Kurian answers, “we’re not talking about a new programming model or a new computing paradigm.

“What we’re talking about is how do you take your existing applications, as well as new modern Internet applications that you may have built, and deploy them at a much lower hardware cost – CPUs, storage, servers – and get the same reli-

“ We’re not talking about grid computing as a paradigm shift in the way people ought to design their applications”

them to write portable applications while continuing to give them the ability to solve more and more relevant problems for their own organizations.

“Some of the vendors,” he says, “try to kill the golden goose, altering the very thing that made their products successful – open standards – by trying to take control of these standards in a proprietary way.”

Web Services – No Cure-All

Larry Ellison’s skepticism about Web services is well known.

“Larry has always said that Web services have an important role to play within IT organizations,” says Kurian, when I ask about Ellison’s perspective. “His skepticism has always been related to when vendors claim that Web services are a panacea for all kinds of IT problems, and that there’s simply a magic whereby if you wrap everything in Web services, they know how to communicate with everything automatically.

“That’s where his skepticism came from,” Kurian explains, “and it’s been justified.

“Our view of Web services at Oracle is that they have a very good value proposition for organizations, particularly in the context of taking and modernizing your existing legacy infrastructures by wrapping them with a more standardized set of interfaces so they can be accessed in a more rational way, a more standardized way, from new systems, new applications you’re building, or even tied into

ability, low cost management, etc., that you enjoyed when you deployed them on big servers.

“We’re not talking about grid computing as a paradigm shift in the way people ought to design their applications. The critical thing, we tell customers, is that any application that you write to a modern architecture like J2EE can be automatically deployed on the grid.

“We are addressing the central question that customers have today,” he adds, “which is that while they have enormous budgetary pressure on their IT infrastructure, they have a lot of excess capacity sitting around in their IT environment.”

Which makes Oracle 10g, presumably, the right app server at the right time?

Kurian smiles. “OAS 10g is the result of over 5 million hours of engineering time, and we spent a lot of time looking at the primary problems in the way they built, deployed, and managed applications. So, yes.”

But his high opinion of 10g isn’t formed in a vacuum.

“Over the last two years we’ve had more than 4 million people download our application server off our Oracle Technology Network,” Kurian notes. “We delivered 10g just before Christmas, in the middle of December, and over the Christmas holidays we had close to 12,000 downloads and the feedback has been great.”

Turning to 2004, I ask Kurian what kind of year he’s expecting, given the givens – election year, a tough economy.

"We are definitely seeing stabilization in the market as well as an uptick in spending," he replies. "The recession has just been a lull in the process of customers transforming themselves from traditional models of business to the Internet-oriented model. That process will continue – people don't argue with the fundamental value of the new computing model.

"In our view," Kurian continues, "the reason we're really excited about 2004 is that in the next two or three years every company in the world is going to go to an Internet model of computing and every company will want to deploy applications on a server, and have the server – the application server and the middleware – provide the capability to do fundamentally new things better, faster, and easier, and to run this with greater reliability and manageability.

"We have been working at this for two years now and we think that this is only going to accelerate now that we are coming out of this tight budgetary environment. What's happened during the recession is that members of our customer community have become much more aware of how they are spending money on information technology and are much more cost-conscious even as they have more dollars to spend."

Pursuing the Dream

Curious as to the source of his continuing, undiminished passion for the technologies he is involved with, I ask Kurian how he explains it.

"It's not just my own passion, it's shared by the great team of people I work with," he points out modestly.

"I think we have one common belief, and that belief is

that this shift of the computing model happens only once every generation," Kurian observes, with more than a hint of awe.

"No one has talked so far about a 'post-Internet' computing model," he continues. In other words, the current cycle is far from over yet.

"Having the chance to help customers and organizations successfully go through that fundamental change is something that very few people get to do, and very few companies get to do. We felt that we wanted to build a software product, an application server, that was so good, so capable, so easy to use and so powerful, that every customer, every company, every organization in the world would try our stuff and like it.

"That's still our dream," Kurian says, "that's still what we work for, and that's still what everyone here at Oracle is focused on. It's just so apparent to us why we feel so passionately about enabling everybody in the world to try our stuff."

Thomas Kurian, in other words, feels almost a sense of personal privilege at being the man he is, in the job he is, at the moment he is.

"I do feel privileged, yes," he replies. "We are all privileged. We are all very lucky at Oracle to have the resources to develop a product that we can make available to so many people. We are privileged that we have 19,000 customers – soon many more hopefully – who use our product.

"It is the greatest privilege in the world to build something that the whole world is going to try, and that continues to be our goal."

With passion like this on tap, it is certain to be an interesting year for Oracle, with or without a PeopleSoft deal. ☺

Your mission depends on having the right tools.



ExtenXLS 3, the 100% Java Reporting Toolkit

With unparalleled support for complex Excel files, ExtenXLS enables creation of feature-rich Excel reports with charts, formatting, text, macros, and formulas, using data from any source.

Best-in-class Excel compatibility, robust features, and a company dedicated to its products.

Download: www.extentech.com/jdixls/

EXTENXLS³
JAVA XLS REPORTING TOOLKIT

extentech™

ExtenXLS, the right tool for complex reporting™

Image AS17-145-22157 provided courtesy of NASA. NASA does not necessarily endorse Extentech Inc. products or service, or companies, trademarks, products or services herein mentioned. Java is a Registered Trademark of Sun Microsystems Inc. Excel is a Registered Trademark of Microsoft Corporation. All other trademarks mentioned herein are the property of their respective owners.

IDE Harvesting

Joe Winchester

Desktop Java Editor

At the opening keynote speech at the EclipseCon conference last month, Erich Gamma showed the Swingset application running inside an Eclipse viewer. For me, it was a definitive demonstration of the two GUI toolkits side by side. Instead of Java GUIs having to choose, it's now mix and match. If you want Swing components go ahead; if you want SWT controls do likewise; and if you want them together in the same window, knock yourself out! Other initiatives like JSR 198, whose goal is to describe a common API for extending IDEs, will benefit from interoperability. I hope I've seen the last newsgroup fan war between people who fight about which toolkit is superior. For those of you who are still arguing, Figure 1 shows an SWT shell with an SWT button on the left and a Swing button on the right. Or is it the other way around?

For everyone who isn't already doing so, let's all focus back on the real job of building killer apps for our users. This segues into my next subject – the buzz at EclipseCon around the rich client project. It's a concept seeded by Todd Williams of Genuitec and others in the community to strip Eclipse bare of all things that are related to its being an IDE, while leaving behind enough of its framework to allow wider-ranging end-user applications to still be built.



Figure 1 SWT and JFC swinging together

In a way it has already begun, because SWT, which was built to be the GUI toolkit for Eclipse, is being used by some developers in their end-user Java applications. The rich client idea, however, is to go even further and allow programmers who are trying to create business applications to benefit from core Eclipse concepts such as perspectives, viewers, editors, toolbars, menus, wizards, and workbench state. I believe there is a huge potential for this to affect desktop software development.

The programmers who created NetBeans, JBuilder, IntelliJ, Eclipse, or whatever your favorite IDE is are some of the most skilled developers in Javaland and have undoubtedly created a whole set of class libraries and solved application problems that are applicable to anyone building a functional Java desktop program. The best frameworks come not from design committees architecting grandiose solutions buried in specware, but instead are harvested from successful real-world applications. These programs have proven their mettle by solving specific prob-

lems and then becoming refactored so that when similar issues are encountered again, the programmers don't have to start from square one and can benefit from the previous efforts. They will reuse the best parts of the original concepts and code and, in turn, provide feedback into the framework so it grows organically, with only the fittest parts surviving.

Michael Tiemann from Red Hat gave a keynote at EclipseCon where he discussed Moore's Law and the industries where it had been successful, as well as lamented why similar productivity gains hadn't occurred with software. In areas where Moore's Law worked, one of the key ingredients was that the output of a process was able to push back into itself, thereby creating an exponential feedback loop; for computer performance, once the transistor was put onto semiconductor programs that could then be written to allow chip designers to design the next generation of hardware, which then helped to speed the design of systems to build the next set of chips, and so forth.

Anyone building a business application for an IT department is unlikely to just start every time with the basic JFC (or SWT) class libraries. Instead they'll create frameworks that deal with navigation, validation, customization, user security, and other areas of programming as it doesn't make commercial sense to reinvent them each time a new department wants something built. I know programmers who, when leaving a company, try to negotiate a deal where they can take their class libraries with them. They view these libraries as part of their value-add that allows them to be productive elsewhere, the same way a skilled craftsman has a favorite set of tools. Some companies even create special framework groups to support their development efforts, although this has its dangers.

Why not have people building line-of-business applications just start with a set of high-level frameworks, and what better place to harvest this than from developers who have created the best-of-breed IDE tools? A colleague of mine once said that the best framework doesn't solve users' problems, rather it allows them to spend more time solving their real issues. In other words, it lets them focus on building their business application instead of on view management or implementing a weak observer pattern.

I hope all of the IDE developers take up this challenge and start to farm their neat stuff back into rich client product offerings. Perhaps then we can begin to achieve Moore's Law for software productivity. ☺

“...the best framework doesn't solve users' problems, rather it allows them to spend more time solving their real issues”

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

joewinchester@sys-con.com



J2EE Web Hosting

<http://www.webappcabaret.com/jdj.jsp>

1.866.256.7973

Quality Web Hosting at a reasonable price...

<JAVA WEB HOSTING AND OUTSOURCING>

You have developed the coolest mission-critical application. Now you need to deploy it. Outsource your hosting and infrastructure requirements with WebAppCabaret so you can save time and money and concentrate on other important things. WebAppCabaret is the leading JAVA J2EE Web Hosting Service Provider. From shared hosting to complex multi-dedicated server hosting, WebAppCabaret has the right solution for you. 30 Day Money Back Guarantee and SLA. WebAppCabaret offers the latest Standards based Servlet containers, EJB servers, and JVMs. We provide options such as e-Commerce, EJB 2.x, Failover, and Clustering. Our Tier 1 Data Center ensures the best in availability and performance.

At WebAppCabaret you have the flexibility to choose the LATEST hosting technology best suited for your WEB application or your programming skill. If you are a programmer or consultant, WebAppCabaret has the right hosting solution for your project or client's dynamic web application/services requirements.

OUTSOURCING:

Do you really need an IT department for your web applications, mail systems, and data backups when we can perform the same functions more efficiently at a fraction of the cost - with competent technical expertise and redundant hosting facilities.

J2EE HOSTING:

Below is a partial price list of our standard hosting plans. (Reseller accounts also available). For more details please log on to <http://www.webappcabaret.com/jdj.jsp>

\$39/mo Enterprise

- Latest JBoss/Tomcat/Jetty
- Latest JSP/Servlets/EJBs
- Private JVM
- Choice of latest JDKs
- Dedicated IP Address
- NGASI Control Panel
- PHP and Perl
- Web Stats
- 1GB Disk
- 200MB DB
- MySql
- PostgreSql
- Dedicated Apache
- Telnet . SSH . FTP
- 5 Domains
- 100 Emails
- Web Mail . POP . IMAP
- more...*

\$191/mo Dedicated

- Managed Dedicated Server starts at \$191 per month for:
- 256MB RAM
- Pentium 4
- 40GB RAID
- Firewall
- Unlimited Domains
- Unlimited Web Site Hosting
- NGASI Control Panel with your own Logo
- Each dedicated server configured for standalone web application and web hosting (resellers) at no extra charge.
- more...*

\$17/mo Professional

- Latest Tomcat/Jetty
- Latest JSP/Servlets
- Private JVM
- Choice of latest JDKs
- NGASI Control Panel
- PHP and Perl
- Web Stats
- 200MB Disk
- 30MB DB
- MySql
- Telnet . SSH . FTP
- 2 Domains
- 20 Emails
- Web Mail . POP . IMAP
- more...*

\$99/mo Reseller

If you cannot afford a Dedicated server for reselling web hosting, for \$99/mo the Shared Reseller plan gives you:

- 5 Professional Plans
- 10 Basic Plans
- NGASI Control Panel with your own Logo
- 50% Discount
- No System Admin
- Easy Account Mgt

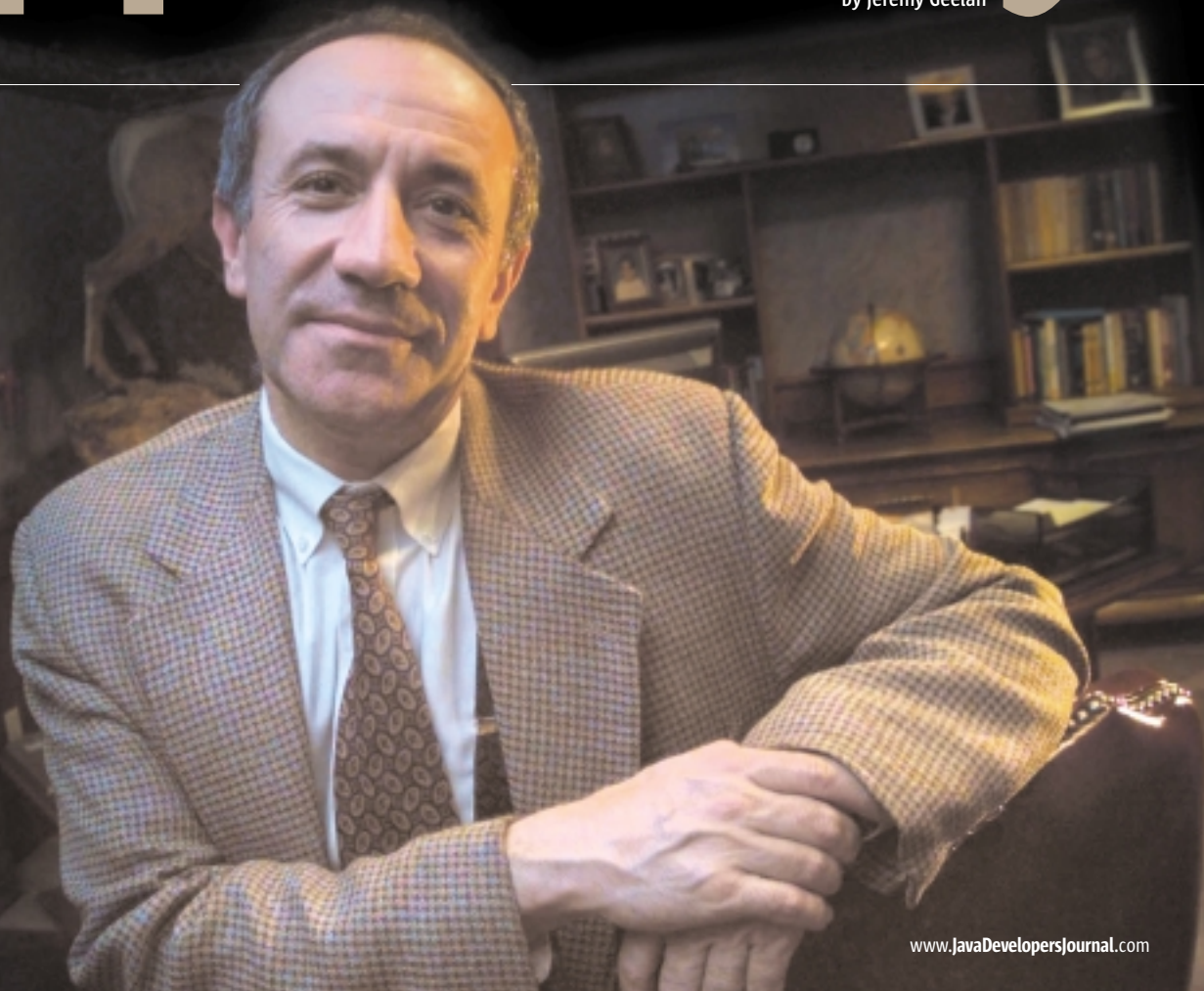
more...

JDJ INDUSTRY PROFILE:

ADAM KOLAWA of Parasoft

‘Java’s Great Missed Opportunity’

by Jeremy Geelan



Successful enterprises and successful business leaders often have two characteristics in common: focus and simplicity. In Adam Kolawa's case – CEO, chairman, and a cofounder of Parasoft since its founding over

15 years ago – the simplicity manifests itself the moment you meet him. Here is a chief executive who knows how to prioritize.

Priority number one, as far as Kolawa is concerned, has always been verification. The alpha and omega of the Parasoft approach is that software cannot be said to work until it is verified – this includes Web services.

“This is something that I learned long ago,” Kolawa explains, “and it’s a major problem within the software industry. Companies build and develop software, hope that it works, but do not go to the necessary lengths to verify that the software actually performs and functions as it should.”

This type of thinking, according to Kolawa, is especially detrimental to the development of Web services.

“Web services are complex systems that attempt to marry software, hardware, new code, legacy code, existing systems, systems in development, and who-knows-what on the client side,” he says. “There are countless opportunities for things to go wrong – a slight mistake in any component or interface will cause problems that ripple throughout the system.”

Kolawa accordingly believes that, as the Web services paradigm grows, managers above all need tools that will allow them to test their Web service initiatives for security, interoperability, and scalability. His own company, Parasoft, has pioneered a methodology known as “Automated Error Prevention,” or AEP, which he believes is the answer to this problem.

What Has Java Done Right?

We will return to AEP later, but now I’m anxious to explore Kolawa’s thoughts on the state of Java – the language, the platform, and the community. What, I wonder, does he think Java has done right...and what not? Is there anything in Java’s history that he wished had turned out differently?

“I think the one thing that Java did right was its portability,” Kolawa replies. “It always compiles and the code is portable at the compilation level. That’s the great thing about Java and is what made it so powerful today.

tinues, “it will die. The main reason why Java came into use was because C++ was too complicated. So it would be wise not to expand the features or make it more complicated than it is.”

What about J2EE, then? Is the latest generation of development tools easing the pain, or does he side with those in the Java community who say that J2EE is too unwieldy and complicated?

“The last generation of development tools did ease the pain somewhat, as they can automate some tedious tasks,” explains Kolawa. “Something else that made development a bit easier is the appearance of the J2EE patterns. There are multiple ways of solving the same problem in J2EE, and that’s where patterns are useful and where they can help a lot.

“Nevertheless, users need to also learn to identify and apply these patterns. And that’s the basic problem with J2EE – it is still too complicated, just the specification is huge. There is a lot to learn before users can start writing any code. Development tools and patterns do help to a certain extent, but they can only do so much. Therefore, a complete knowledge and understanding of J2EE concepts would ease the pain more than any tool.

“Also, when things go wrong with the program it is quite difficult to find out what happened and to make things work correctly in multiple application servers.”

What about Java on the desktop? Has Java lost the desktop battle to the likes of Flash and .NET?

“Yes, I think Java has lost the battle because of three major problems,” Kolawa observes. “The first problem is that the Java Virtual Machine isn’t being distributed with desktops, and to make an application reliable, you have to bundle it with the Java Virtual Machine. However, this creates the dilemma of large downloads,” he adds.

“The second problem is that applications written in Java simply do not look as nice as those written in Flash or .NET. And finally, Java is slow on the desktop. Although this doesn’t matter for servers, it is obvious that Java’s speed hurts it on the desktop.”

And does the JCP work? Is the principle sound or flawed?

“Well, every structured committee has its benefits and advantages depending on how well it is organized. In my opinion, the JCP can be successful if the right people are involved and there is strong leadership. However, if the decision makers of the JCP are interested only in manipulating it to their advantage, then obviously the JCP will be less successful due to poor leadership,” Kolawa replies prudently.

“Linux and Java go perfectly hand-in-hand. If it weren’t for Linux, Java would not be as successful as it is, and if not for Java, Linux would not be as successful either. It’s a perfect partnership”

“The single thing that I would like to change about Java’s history is its separation from Microsoft. I think it is a shame that the technologies from both sides cannot be used together. Java seemed to be the perfect technology to bring Microsoft into the world of coexistence, and I think this opportunity got lost.”

What, then, does Kolawa think the future holds in store for Java?

“I would be sure not to allow further expansion or complication of Java,” he says immediately. “In order for it to stay successful, it needs to stay simple.

“If we start adding new features and syntaxes,” Kolawa con-

Java Startup?

Next we play a brief game. We imagine for a moment I have some serious money to invest in a Java startup company. What type of company would he advise me to put my hard-earned dollars into? What’s going to be big in the next 18 months?

“First of all,” Kolawa says, candidly, “I would advise you to not invest in a Java startup company!”

“There are already big players in this market,” he adds, “and more and more are coming into the picture. It’s going to be more crowded and difficult for a new startup to break through and be successful.”

And even if it were his own money, no difference? He'd still look outside of Java?

"Under no circumstances would I invest in a Java startup company. If I do invest any money, I'll be sure to invest it in another type of company."

Web Services – What's the Downside?

Next we take a critical look at the Web services paradigm. Does Kolawa believe the view that it's really the philosopher's stone, the answer to all of an enterprise's integration problems? What are the consequences of the trend in enterprises to build fewer and fewer pure new applications and instead repurpose what they already have? Is it a poisoned chalice, I ask him – condemning us forever to be seeking the Holy Grail of integration?

"No, I don't think we will be condemned to search for any sort of Holy Grail," Kolawa replies. "I think that building fewer and fewer pure new applications is good in any industry that is trying to mature."

"Web services help us connect existing applications that are already running well," he adds. "This saves companies

Why Test

In return for such straight talking, it seems only fair to ask Kolawa about the subject of Parasoft, and about his vision of Automated Error Prevention software solutions that can help companies eliminate errors in software development throughout the world.

How would he define what inevitably is now called "AEP"?

"Put simply, AEP is the common-sense approach to controlling errors in software, and verifying that software performs and functions as intended," Kolawa replies. "As companies and consumers rely more on Web services, the importance of quality and reliability will continue to increase, thereby increasing the importance of Web services verification – including the verification of security, interoperability, and scalability."

Knowing that SOAPtest 2.5 is something Parasoft has just launched, I ask Kolawa whether there is a competitor to SOAPtest 2.5 – and if not, why not?

"While there are other Web services testing tools on the market, none of them are close to being a competitor to SOAPtest 2.5," he says. "As a part of Parasoft's AEP methodology, SOAPtest is the most comprehensive tool for Web services, allowing users to verify all aspects of a Web service, from WSDL validation, to unit and functional testing of the client and server, to performance testing."

It is SOAPtest's "flexible nature," according to Kolawa, that makes it an ideal choice for development engineers and QA professionals alike.

"As an important part of Parasoft's AEP methodology," he explains, "SOAPtest provides extended benefits when used in a team environment."

The Parasoft claim then is that it's the only Web services tool that supports an enterprise's entire Web services development team?

"Precisely. It's fully configurable for use throughout your Web services development group, allowing the exchange of development and test information between individuals and groups."

SOAPtest also provides individual and project-level reporting for easy test result distribution among group members, Kolawa continues.

"In addition," he adds, "SOAPtest can also be used in conjunction with other Parasoft tools to build a system-wide verification process."

I ask for an example. "DataRecon can be used to verify the many problems related to SQL injections in Web services," Kolawa says. "Also, JTest and .Test can be used to test and verify Java and .NET implementations of Web services. Using SOAPtest by itself, or with other Parasoft tools, users can be certain that their Web services are fully functional from conception to deployment."

So Parasoft not only provides the tools, he notes, but also the methodology necessary to achieve superior development of Web services.

time and allows them to free up other resources to build and develop things that are needed at a more immediate time.

"Normally, organizations are so overwhelmed with things that technologies like Web services allow them to move forward, not backward. The more businesses develop from pre-configured parts the better. Just because developers repurpose and build from what they already have doesn't necessarily mean that new technologies will cease."

"Developers will always be developing new applications for new purposes," he says with characteristic optimism, "and there are constantly new ways in which we can pour human intelligence into computers."

'SOA Will Succeed'

What about Kolawa's take on service-oriented architecture – is SOA just the new buzzword du jour, or does it have substance?

He answers without hesitation. "I think that service-oriented architecture is going to move forward and be successful," he says quite firmly. "There definitely is substance behind SOAs and this can be seen in the development in Web services such as e-Bay and Google, for example."

"These companies are starting to expose their APIs so consumers can take advantage of these available services. It's like I just said, companies like this are building from preconfigured systems, but Web services still allow for new opportunities. People are connecting their inventories to e-Bay and selling more products through this. Opportunities like this were never thought of, and now they are possible because of SOA."

Last, it wouldn't be right to pass up on the opportunity to ask Kolawa what he thinks the overall effect on Java will be of the compelling economics of Linux.

"Java has helped Linux so much because Linux is such a good engine to run Java middleware," Kolawa explains. "Linux and Java go perfectly hand-in-hand. If it weren't for Linux, Java wouldn't be as successful as it is, and if not for Java, Linux wouldn't be as successful either. It's a perfect partnership."

In Kolawa's view the hearts of enterprise systems will continue to be built on these technologies because they are relatively inexpensive.

"Java already has large amounts of already-written legacy code and has an extensive library," he says, "and Linux has proven to be very secure. In my opinion, I think these two technologies will dominate the middleware market, and I don't think there's room for .NET to break in."

Does he think open source is The Big Next Thing for the computing world?

"Yes and no," Kolawa answers carefully.

"Some things, like operating systems, should no longer be proprietary," he continues. "It's an old technology and it's well known how they are built."

"But somebody has to pay for innovation, so we will have a mix of proprietary and open source products. New software must be built by companies, and that will continue to be sold proprietarily. But after a while, some things become too obvious and they will move into the public domain."

Kolawa ends, as he began, with a simple point – though not everyone will necessarily agree with him (Darl McBride for one!).

"Software is like classical music," he says. "For instance, when a composer writes his music, it belongs to him in the beginning stages. But after years and years of playing this music for audiences, the song becomes familiar, and it becomes public property where anyone can play it."

I can't help wondering what the litigious SCO Group would have to say to that! ☺

LOOK FOR YOUR FREE...

IT solutions >GUIDE

»Linux »Java »Web Services ».NET »XML »Wireless »Storage »Security

The Premier Resource for Today's Corporate & IT Decision Makers

VOL 1 ISSUE 1 SPRING 2004

WWW.SYS-CON.COM/IT

TECHNOLOGIES YOU NEED NOW!

How to Manage Your Ideas Using Today's i-Technologies

- » Delivering Software as Service
- » Leveraging Linux/Open Source
- » Moving to a Service-Oriented Architecture
- » Desktop Software: Migrating from Server to Client
- » Using Developer Tools to Drive Cost Out of Software
- » Application Integration
- » Storage & Security

Reaching 135,000 Corporate and IT Decision Makers

© 2004 SYS-CON MEDIA. ALL RIGHT RESERVED. ALL BRAND AND PRODUCT NAMES USED ARE TRADE NAMES, SERVICE MARKS, OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

Coming this **SPRING!**

Java Technology on Mac OS X



COFFEE WITH YOUR APPLE PIE?

by Daniel H. Steinberg

Let's create a wish list of items you'd like to see in a Java-friendly operating system. For development you need a robust operating system that makes it easy for you to code, compile, debug, build, and deploy your Java app.

It would be nice to have a choice of text editors or IDEs. Your favorite open source tools should work well and the OS shouldn't get in your way. For deployment, a solid implementation of a recent JRE is essential. It would be even better if you could count on a particular VM being installed and tuned for your platform. There are 10 million copies of Mac OS X with Java preinstalled on a Unix box with an end user-friendly front end. Believe it or not, the platform of choice for developing and running Java applications is the Panther edition of Mac OS X.



Panther for Java Developers

You may think of the Mac as a machine for your mother or maybe for the local kids. You are sure that serious Java developers would never be caught using a Mac – or would they? Over the last year or so you may have noticed more and more Apple PowerBooks at developer events. From JavaOne to OSCon to ApacheCon, there are considerably more hard-core alphas Mac users than the 2% of the market share reported in the popular media for the wider audience.

The Mac has many of your favorite geek tools pre-installed with user-friendly interfaces added. Your mom will find that she can turn on Apache Web Server, which comes standard on Mac OS X, by checking a box. If you're so inclined you can open up a terminal window and access the command line to enter your favorite Unix commands. All of your favorite hard-core tools are included. You can write code using Emacs or Vim and check it in using the standard CVS installation. You can write in Perl, Python, PHP, and Ruby. You can install the free developer tools and

use Apple's XCode to write platform-specific code in Objective C, C, or AppleScript. You can also run your favorite X11 applications using the X11 for Mac OS X implementation of the X Window System. In addition, Mac OS X has always shipped with J2SE 1.3.1 and now includes J2SE 1.4.x as well.

James Gosling wrote in his java.net blog:

I use the Mac because it's a great platform. One of the nice things about developing in Java on the Mac is that you get to develop on a lovely machine, but you don't cut yourself off from deploying on other platforms. It's a fast and easy platform to develop on. Rock solid. I never reboot my machine... Really! Opening and closing the lid on a PowerBook actually works. The machine is up and running instantly when you open it up. No viruses. Great UI. All the Java tools work here: NetBeans and JEdit are the ones I use most. I tend to think of OS X as Linux with QA and taste.

Many of your favorite open source applications are easily installed on Mac OS X. In the server edition of Panther you'll also find Tomcat, JBoss, and MySQL preinstalled with tools for managing them remotely. The versions that ship with Panther and Panther Server may not be the most recent. Installing Tomcat, for example, is trivial. Getting it up and running is a matter of setting the JAVA_HOME variable in the catalina.sh shell script. Applications such as WebLogic and WebSphere are not qualified on Mac OS X and yet they run fine once you get them installed and configured. Because the Mac is, at heart, a Unix box and because the J2EE stack runs on top of the J2SE stack, the most difficult part of getting your favorite app servers or other enterprise tools running is setting variables and tweaking the installation process to put files in the Mac-appropriate locations.

Not Like the Bad Old Days

The Mac has not always been a good platform for developers. If you tried programming on a Mac under System 7 or System 8, you may not have fond memories of the experience. Often you were able to get in only a compile or two before your system froze and you were forced to reboot. The most optimistic could look at this as a chance to stretch their legs and get another cup of coffee. Often it was five minutes of waiting for the machine to come back up and another ten trying to get your head back into what you were doing.

Daniel H. Steinberg is the editor of ONJava and java.net. He's been working with Java on the Mac since it first appeared but also enjoys coding in ObjC and other languages. He is a long-time technical writer, trainer, and developer with Dim Sum Thinking, Inc., and recently coauthored the book *Extreme Software Engineering: A Hands-on Approach*.

daniel@oreilly.com

On top of all of that, working with Java on the Mac presented its own set of problems. First, each new J2SE was released on Windows at least a year before it was available on the Mac. Second, the performance of Java applications on classic Macs was never very good and the applet performance was tied to increasingly out-of-date browser plugins. The presence of multiple JVMs on the Mac platform improved the situation slightly in the short run but represented a great deal of duplicated effort that made it a bad long-term strategy. Third, Sun and Apple did not see eye to eye on the direction and implementation of Java. As Steve Jobs said at the JavaOne 2000 keynote when Scott McNealy asked what had taken so long to ship J2SE along with the Mac operating system, “You guys were busy putting Java in light bulbs.” In addition, many developers were focused on Java in the enterprise and this did not fit Apple’s strategy at the time.

The lag between the J2SE release for platforms Sun was targeting and the Mac release has long been an issue. Each time Apple has announced that the extra time taken on the previous port should pay off by reducing this lag in future releases. Historically that has not been the case. Most recently, however, the delays in the J2SE 1.4 release were the result of porting the GUI from Apple’s Carbon framework to their Cocoa framework. This Cocoa port helps your Java application look and feel more like a native application even before you tune it, and it allows you to take advantage of Apple functionality. This change initially broke Quicktime Java (QTJ) on Mac OS X and resulted in QTJ applications working on Windows and not on Apple’s own platform. More recent versions of QTJ work on Mac OS X but do not present all of the functionality present in earlier versions.

Develop on and Deploy to Mac OS X

J2SE 1.4.1 ships as part of Panther, and now Java 3D and JAI are available. Performance issues are still reported by those deploying their applications on multiple platforms. Often those reporting apparent performance problems to the Java-Dev list (available at <http://lists.apple.com>) are given advice on how to address their particular issues. The list is a friendly place as long as you don’t ask “When will Apple release <fill in what you’re waiting for here>?” Panther Server ships with Tomcat and JBoss, and in the last couple of years WebObjects

has been rewritten as a very Java-friendly platform for enterprise development.

Interaction among Apple developers tends to be friendly and helpful. At the same time, there have been improvements in Apple’s relationship with Java developers. Instead of making developers wait for FCS Java releases to experiment with Apple’s implementation, ADC (Apple Developer Connection) members were given early releases of Java 1.4. This enabled developers to beat up on the beta releases and help identify bugs on the way to releasing Java 1.4 to end users as part of Software Update. A smaller group was able to work with the QTJ APIs and help shape the most recent release. ADC members have had access to prerelease versions of J2SE 1.4.2, JAI (Java Advanced Imaging), Java 3D, and others. You can become a member of the ADC for free at <http://connect.apple.com>.

The release of JAI and Java 3D indicate that the political climate has changed as well. Now that Mac OS X sits on top of BSD Unix, porting Java technology that works on Solaris to the Mac is often not the technical challenge that it once was. The challenge centers on licensing and getting agreements from two companies with different corporate cultures and business aims. With other technologies the issues are technical, not political.

The Eclipse team has recently brought their SWT framework to the Mac. While Apple engineers were busy moving the Swing stack from Carbon to Cocoa, the Eclipse engineers decided to build SWT on top of Carbon. Their public explanation was that they saw SWT as being a peer of Cocoa that fixed many of the problems they had been solving in different and incompatible ways.



Summary

In short, Mac OS X on a PowerBook gives you a powerful and portable Unix server. Whether you use a laptop or desktop, a G5 dual, G4, or G3, you will have all of the hacker-friendly tools you would expect along with a GUI layer that makes the Mac an ideal target platform for your applications. You can’t deploy a JVM along with your Java applications, which makes the download smaller and the installation easier. You’ll find helpful articles on tuning your Java application for Mac OS X online at www.macdevcenter.com and at <http://java.sun.com/developer/technicalArticles/JavaLP/>. ☺



From JavaOne to OSCon to ApacheCon, there are considerably more hard-core alphasgeek Mac users than the 2% of the market share reported in the popular media for the wider audience”

IntelliJ IDEA 4.0

by JetBrains, Inc.

Reviewed by
Vijay Phagura

The integrated development environment (IDE) is a big part of the toolkit of a modern software developer. Java projects have complex development processes, especially if the project is going to be developed using open source and J2EE technologies. A smart and efficient IDE plays an important role in making developers more efficient and productive in doing their tasks and meeting project deadlines. This review discusses one such smart IDE – IDEA 4.0.

JetBrains, Inc., develops the IntelliJ line of products and IDEA is their IDE for Java development. At the time of this writing, IDEA 4.0 was on its way to being released.

IDEA is not only a server-side development IDE, the new release also has a GUI designer for front-end development and design. IDEA integrates various tools like Ant, JUnit, and CVS. It provides a lot of help to the developer without getting in the way. The keys, editor settings, SDK settings, path, etc., are fully and easily configurable.

An evaluation copy can be downloaded from their Web site with your choice of platform, and installation is easy and smooth. I'm sure you'll be very pleased with its documentation, which is well written and extremely helpful to developers.

The following is a brief description of some of IDEA's features, most of which are important for any Java developer.

Those Import Statements!

If you're a developer who has been doing Java for some time, you'll love this one – IDEA has a feature called Import Assistant. As soon as you use a class name that's not imported in your current class, it'll pop up a hint showing the qualified package name of the class you just typed in and a help tip showing Alt-Enter.

As shown in Figure 1, if you press the Alt-Enter keys the import statement relevant to the class you are typing in will be inserted. Also, when a chunk of code is pasted from some other class, it will pop up a list of imports that you'll need. By just pressing Alt-Enter, all the imports are in your class.

IDEA also indicates to the developer any irrelevant imports in a class by graying it out, thus optimizing the import statements.

Do I Have to Write Another Loop?

Tired of writing iterations and loops? There is a very powerful feature called Live Templates. If you want to write some code to iterate through a List, just type itli, press Ctrl-J, and the entire List iteration code will be written for you (see Figure 2).

Of course, you can go in and change whatever you want, but the best part is that the most tedious work is eliminated

from the developer's list. This is a big time-saver.

There are many other templates available in this IDE that help code the most commonly used code sequences. For instance, `System.out.println()` – you don't have to type the complete statement over and over again.

IDEA also allows you to write your own templates and store them. Another feature, somewhat related to this one that's worth mentioning, is that IDEA provides the developer with an option to surround a piece of code with `if/else`, `try/catch`, `while`, etc.

Refactoring

IDEA does a good job of refactoring. It provides a developer with most of the refactorings like renaming and signature changes. The process of refactoring has been simplified and IDEA helps the developer do a selective refactoring too. To make refactoring work properly, the user is given only a "Save All" option to save his or her files so that the developer does not omit any changed files while saving.

More Features

The features described above gave you a flavor of what IDEA 4.0 has to offer. There are many more helpful features supported, such as code reformatting, searching, and inten-

```

version = Integer.parseInt(root.getAttribute()),
}
catch (NumberFormatException e) {
    java.io.File? Alt+Enter
    File file =;
    return version;
}

```

Figure 1 Import assistant

```

46 button = new JButton("");
47 button.addActionListener(this);
48 pane.add(button, BorderLayout.WEST);
49 setSize(100, 100);
50
51
52 itli Iterate elements of java.util.List
53 itco Iterate elements of java.util.Collection
54 iten Iterate java.util.Enumeration
55 itit Iterate java.util.Iterator
56 itli Iterate elements of java.util.List
57 ittok Iterate tokens from String
58
59 itve Iterate elements of java.util.Vector

```

Figure 2 Using short cut for Live Template

Vijay Phagura, a professional Java/J2EE consultant, has over 12 years of experience in software architecture and development. He specializes in designing and developing software using J2EE and other Java technologies.

vphagura@yahoo.com

JetBrains s.r.o.

Klanova 9/506
14700 Prague
Czech Republic
Phone: +420 2 4172 2501
Fax: +420 2 6171 1724
E-mail: info@intellij.com
Web: www.jetbrains.com

Specifications

Platforms: Any platform with JDK 1.4.x

Test Environments

Test Machine 1: Win 2000, 500MHz CPU, 128MB RAM
Test Machine 2: Win XP Pro, 3GHz CPU, 512MB RAM

Snapshot

Target Audience: Java developers and architects

Level: Beginner to advanced

Pros:

- Powerful code completion
- Powerful refactoring
- Powerful editor
- Clean tool integration
- Easy to use
- Fully configurable
- Efficient memory management
- Good documentation

Con:

- Difficult to import existing EJB projects

Summary

If you're a Java developer, then IDEA is for you. It takes a lot of repetitive, boring, and time-consuming code off your list so you can spend more time designing, optimizing, and improving the product quality. It provides a total integrated environment for Web and J2EE development. A developer does not have to use different editors or tools for XML, JSP, etc., as they're all integrated in this one single tool. For convenience and usability, it cleanly integrates most of the popular industry tools like Ant, JUnit, and CVS.

I used this IDE on at least two different projects: one with EJB2.0 and the other a Web application. I used it on two machines and it performed well on both machines and projects. After you get used to it, Java development is a breeze! ☺

tion actions. You can read about these features on the JetBrains documentation site.

Web as well as EJB development is well supported by IDEA 4.0, and it now supports BEA WebLogic deployment. I've tested both of these environments and they're great to use. For instance, if your J2EE project supports role-based security, using IDEA will greatly simplify modifying the ejb-jar.xml.

Also, it includes generics support. This is a useful feature for supporting Java 1.5.

ODTUG
NOW

2004

LET US SHOW YOU HOW

- To build a J2EE application with an Oracle database
- To integrate web services and Java components into Forms
- To utilize 9i/10g database features in application development
- To know when to use the various application development options
- To get the most out of Designer, Developer, and Warehouse Builder

ORACLE DEVELOPMENT TOOLS USER GROUP

Preconference Business Rules Symposium—Saturday, June 19
JUNE 20-24, 2004 THE WESTIN KIERLAND RESORT & SPA SCOTTSDALE, ARIZONA www.odtug.com



Onno Kluyt

From Within the Java Community Process Program

The JVM, J2SE 1.5, and more

Welcome to the March edition of the JCP column. Each month you can read about the Java Community Process: newly submitted JSRs, new draft specs, Java APIs that were finalized, and other news from the JCP. This month I'll discuss the public review of the upcoming J2SE 1.5 technology and two JSRs that are part of that release, Nokia's JSR 226, and a JSR in maintenance review.

J2SE 1.5 in Public Review

After completing its Community Review, JSR 176, the so-called umbrella JSR for the J2SE 1.5 release, quickly entered into Public Review. In parallel, a beta release of the reference implementation was also brought out so that there are not just draft specs to look at but real software to play with. J2SE 1.5 features many great new items, including the various language functionalities such as generics, metadata, and auto-boxing. I want to mention a different one though – VM sharing – a new feature in the beta release. This is a complex functionality and the J2SE engineering team is keenly interested to find out from you how this is working out, so if you could share your experiences with them? Thanks!

JSRs being run so well. This JSR managed to find a balance between keeping the working group small enough so that it can get work done and making the thought processes of the working group accessible to whoever was interested.

The technology that JSR 166 covers is very specific, and the spec lead knew that there were only a few recognized experts on the topic who also had the ability to commit substantial time each week to work on this, and that there were many more who may not have had the time or the same level of expertise but were very interested in its progress nonetheless. Some of the tools that the expert group used to achieve this were multiple mailing lists, a repository to track changes to the draft specification, and the frequent release of intermediate drafts to a large audience of observers.

JSR 226

In the J2ME environment this JSR, led by Nokia, just recently successfully navigated the Community Review stage. This proposal works on a Scalable 2D Graphics Vector API, and it depends upon the W3C's Scalable Vector Graphics (SVG) format. Examples of the use of this API are map visualization and scalable icons. This API is aimed at the J2ME MIDP

“ JSR 166... managed to find a balance between keeping the working group small enough so that it can get work done and making the thought processes of the working group accessible to whoever was interested ”

JSR 133

JSR 133 entered into Public Review together with the umbrella JSR. This JSR updates the memory model and threading specification for the Java Virtual Machine and the programming language. The goals of this JSR are to ensure that developers can write reliable multithreaded applications while understanding which thread constructions are legal, and to enable virtual machine implementers to build this technology in a high-performant way.

Openness and Transparency

JSR 166 works on a set of concurrency utilities and is led by Doug Lea. This specification effort is part of J2SE 1.5, but I included it in my column for a different reason. The manner in which this JSR is organized is a great illustration of what is actually possible within the rules of the JCP process. JCP 2.6, which will be launched in March, provides best practices for most of its processes and hopefully will result in many more

2.0 platform. The JSR has relationships with several other J2ME-based JSRs, such as JSR 135 (Mobile Media API), JSR 184 (Mobile 3D Graphics), JSR 209 (Advanced Graphics and User Interface), and JSR 217 (Personal Basis Profile).

Maintenance Review

After a JSR goes final, the story doesn't end there, of course. Developers start working with the technology, questions come in, clarifications and new features are requested, and so on. Typically one of two things happens next. Either the spec lead starts a new JSR to develop a major new version of the API or the spec lead (now called maintenance lead) proposes a set of smaller changes, clarifications, corrections, etc., to the community. This is called the Maintenance Review stage and JSR 115, the Authorization Contract for Containers API, is currently in this stage.

That's it for this month. I'm very interested in your feedback. Please e-mail me with your comments, questions, and suggestions. ☺

Onno Kluyt is the director of the JCP Program Management Office, Sun Microsystems.

onno@jcp.org



DESKTOP

CORE

ENTERPRISE

HOME

Advertiser	URL	Phone	Page
Altova	www.altova.com	978-816-1600	23
BEA Systems	http://dev2dev.bea.com/workshop1	800-817-4BEA	Cover II
Borland	www.go.borland.com/j3	831-431-1000	9
Crystal Decisions	www.businessobjects.com/v10/047	800-877-2340	17
DataDirect	www.datadirect.com/jdj	800-876-3101	11
Dice	www.dice.com	877-386-3323	33
Extentech	www.extentech.com/jdj/	415-759-5292	31, 51
Google	www.google.com/cacm		27
GreenPoint	www.webcharts3d.com/demo	212-765-6982	35
ILOG	jviews-info-kit.ilog.com	1-800-for-ILOG	25
InetSoft Technology Corp.	www.inetsoft.com/jdj	888-216-2353	43
Information Storage & Security Journal	www.sys-con.com.com	888-303-5282	63
InterSystems	www.intersystems.com/match3	617-621-0600	4
IT Solutions	www.sys-con.com	888-303-5282	57
JadeLiquid Software	www.webrenderer.com	61 3 6226 6274	7
Java Developer's Journal Resource CD	www.sys-con.com/freecd	888-303-5282	65
Mercury Interactive	www.mercuryinteractive.com/optimizejee	800-837-8911	15
NetIQ	www.netiq.com/solutions/web		29
Northwoods Software	www.nwoods.com/go	800-434-9820	41
Oak Grove Systems	www.oakgrovesystems.com/jdj	818-440-1234	47
Oracle Development Tools User Group	www.odtug.com	910-452-7444	61
Parasoft Corporation	www.parasoft.com/jdj_02	888-305-0041	21
Quest Software, Inc.	http://www.quest.com/jdj	800-663-4723	Cover IV
ReportingEngines	www.reportingengines.com	913-851-2200	3
SESMA	www.sesma.com	949-559-SESMA	37
Software FX	www.softwarefx.com	800-392-4278	Cover III
WebAppCabaret	www.webappcabaret.com/jdj.jsp	831-464-6941	53
Wily Technology	www.wilytech.com	888-GET WILY	39

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.

**PREVIEWING MAY 11, 2004,
AT NETWORLD+INTEROP, LAS VEGAS**

INFORMATION STORAGE + SECURITY JOURNAL!



**FOR MORE INFORMATION VISIT
WWW.ISSJOURNAL.COM**



From the World's Leading i-Technology Publisher

©COPYRIGHT 2004 SYS-CON MEDIA. ALL BRAND AND PRODUCT NAMES ARE TRADE NAMES. SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

JSF: The Ultimate in Flexibility? Or Complexity?



Steve Benfield

I have a love/hate relationship with J2EE. I love the idea of standards that we can all use in our development to improve interoperability, ease integration issues, create a pool of skilled developers, etc. I hate the idea that I have to wait years for the standards to evolve and become usable. And I hate having specs that seem to work well in theory but have no practical implementation behind them. This brings me to the JSF specification.

How long have we heard about JSF (JavaServer Faces) and how it will make it easier to build Web pages? Why did it take more than two and a half years to come up with JSF, which is essentially an event model for Web pages? And it happened at a time when Microsoft was coming out with major innovations (a debatable point, I concede) and huge pushes targeted at providing ease of development for the corporate developer. Visual Studio/.NET developers look at the J2EE community and basically say “these guys don’t get it.”

Let’s take a look back at JSPs. Isn’t one of J2EE’s goals to encourage encapsulation and object-orientation? If so, why create a standard that encourages a mixture of presentation logic, application logic, business logic, and data access into one page class? The answer is Microsoft. It was a reaction to what Microsoft had: “Hmm, we need something to compete with ASP. Hey, let’s make JSP. We’ll have tag libraries so we can easily embed components on a page and call it encapsulated.” Obviously things have evolved a bit since then and we can now separate the flow of the application from the pages with frameworks like

Struts. Well, last time I looked, Struts isn’t part of J2EE – which means you’re doing nonstandard development – but it is open and has an ecosystem of developers so it’s a pretty safe choice. Of course, if you like editing XML, Struts is for you. There are Struts editors out there to help you, but in the end you’re still dealing with a lot of complexity.

Now we have JSF, which promises to apply the “ease” of

Swing to page development. My expertise is in building corporate business applications and the needs of the business application developer.

What I see in JSF is overengineering. Let’s think of every possible usage pattern for Web pages and address them, not necessarily a bad thing. The resulting standard is one in which there are several layers of abstraction, many, many moving parts, and ultimate flexibility.

The problem is that ultimate flexibility comes at a cost – it’s called ultimate complexity. From what I can tell, JSF won’t actually reduce the complexity of creating Web applications. In fact, it greatly adds to the complexity. Just as JSPs did too little to provide granularity in building pages, JSF goes too far and makes things too granular. Do I really need a separate event to be called for each piece of data that has changed on a form? And possibly separate handlers for each? Be prepared for the framework that will sit in front of JSF to simplify its complexity.

In defense of JSF, it does follow many of the object-oriented Web page design concepts that we had at SilverStream in, oh, 1998. And it does have some appeal to the PowerBuilder developer that I used to be in

the mid-’90s. However, experience has shown that building page-based applications with the level of object granularity shown in JSF doesn’t make life easier; it makes it more complicated. I’ve architected and used many frameworks in my career and the trick is to perfect the balance between the ultimate flexibility that the architect wants and the ultimate usability by nonengineer developers.

I just joined ClearNova, and for several years our product, ThinkCAP, has provided a rapid way to build applications and presentations, one thing that attracted me to the company.

We’re doing the R&D needed to move parts of the platform toward

JSF, but I’m concerned we’ll lose a lot of the productivity that our current product has by implementing a spec that only object-oriented theorists could love. If we don’t allow developers to have complete control over everything that JSF can do, we’ll take flak from the purists, even if doing so makes the tool so hard to use that it appeals only to experienced J2EE developers.

Ultimately, my fear is that JSF will once again play into the hands of Microsoft’s “complexity attack” and tempt business developers to run for the J2EE exit. ☺



Steve Benfield is VP, Business Development and Technology Evangelism at ClearNova. Previously he was CTO of SilverStream Software. Steve was the editor of SYS-CON’s first magazine, *PBD*, in 1994. stevebenfield@sys-con.com

FREE* CD! (\$198.00 VALUE!)

Secrets of the Java Masters

Every *JDJ* Article on One CD!



— The Complete Works —

CD is edited by *JDJ* Editor-in-Chief Alan Williamson and organized into 33 chapters containing more than 1500 exclusive *JDJ* articles!

All in an easy-to-navigate HTML format! **BONUS: Full source code included!**

ORDER AT WWW.SYS-CON.COM/FREECD

**PLUS \$9.95 SHIPPING AND PROCESSING (U.S. ONLY)*

©COPYRIGHT 2004 SYS-CON MEDIA. WHILE SUPPLIES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE. ALL BRAND AND PRODUCT NAMES ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.



Only from the World's Leading i-Technology Publisher



Jack Martin

Sarbanes-Oxley: The New Rising Star

Ineffectual corporate management has given a great gift to programmers, system administrators, and CIOs – endless corporate accounting scandals. Our federal government has not missed this scandalous behavior as they have passed an extraordinarily strong, far-reaching law to contend with financial fraud.

Officially it's called the Public Company Accounting Reform and Investor Protection Act, but it's more commonly known as the Sarbanes-Oxley Act. The Sarbanes-Oxley Act is a very special piece of legislation designed to keep corporate managers honest. The Act awards dishonesty with a 10- or 20-year prison sentence for CEOs and CFOs. In addition, it has a provision for taking any and all ill-gotten gains from the dishonest executive. There are approximately 14,000 publicly traded companies in the United States, with just about 7,400 of them traded on the New York, the American, or the NASDAQ stock exchanges.

Most recently, the Securities and Exchange Commission (SEC) charged Jeffrey Skilling, the former president and chief executive officer of Enron, with fraud. The SEC is seeking to seize all of his ill-gotten gains and permanently bar him from acting as a director or officer of any publicly held company. On top of that, he is facing a maximum of 325 years in prison and hundreds of millions of dollars in fines. Considering that Andrew Fastow, who reported to Skilling at Enron, got off with 10 years in prison and forfeited \$23 million in cash and assets, this is serious stuff!

Executives are now directly responsible for establishing and maintaining an adequate internal control structure and procedures for financial reporting. The reports must contain a written assessment of the effectiveness of the internal control structure and procedures of the issuer with regard to financial reporting, as of the end of the most recent fiscal year of the issuer.

To put this in perspective, the average billion-dollar company has about 50 disparate financial systems up and running at any given time, some of which have been running for more than 30 years. Remember Y2K, with all of those old systems that were designed in the 50 years after World War II that did not anticipate ever needing to factor in the century changing? Starting June 15, 2004, you need to ensure that everything coming out of the old mainframes, client/servers, and new application servers is correct and adds up perfectly.

Adding to the complexity of the corporate, information-technology topology is the fact that most of them are running two or three enterprise resource planning (ERP) systems. These systems take literally years to install and configure and not a single soul on Earth can guarantee that any ERP system is 100% on the money 100% of the time.

Putting the icing on the cake: about 50% of the time, this information is output to an Excel spreadsheet, opening the door for endless misinterpretations and unintentional mistakes.

Sarbanes-Oxley compliance is very different from Year 2000 readiness. With the Y2K fire drill all you needed to do to achieve success was get the computer to roll over on January 1, 2000, and not shut off or miscalculate. In addition, Y2K was a one-time event. As I'm sure you know, there was no Y2.1K. Sarbanes-Oxley compliance is an everyday, every hour issue that must all be rolled up into a tight, neat package every three months to support the quarterly financial statements.

For a public company to comply with Sarbanes-Oxley, the accounting, financial management, and legal departments all ultimately funnel their data through information technology, which opens issues as to the completeness and accuracy of every code fragment and algorithm that's embedded in any project.

White-shoe law firms, the large accounting firms, and a myriad of management consultants have all begun Sarbanes-Oxley practices. In Silicon Valley and the other technology hot

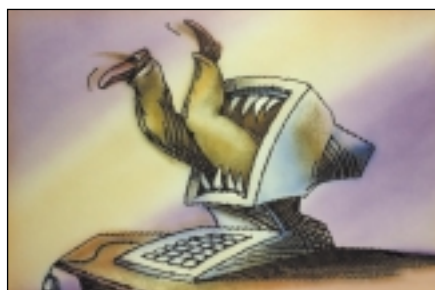
spots around the U.S., venture-funded Sarbanes-Oxley software companies are beginning to appear with increasing regularity.

A good example is Nth Orbit, which is offering a Sarbanes-Oxley product called Certus that provides a systematic approach to compliance. Their lead investor is Sequoia Capital. Sequoia was an early investor in and worked with Cisco Systems, Yahoo!, Redback

Networks, Google, Network Appliance, Cypress Semiconductor, Vitesse Semiconductor, Apple Computer, and Oracle. These are not stupid people! Merger and acquisition activity is also beginning in this space, exemplified by EMC's recent purchase of Documentum. And there is an entire magazine dedicated to Sarbanes-Oxley compliance – the *Sarbanes-Oxley Compliance Journal* (www.s-ox.com).

The Sarbanes-Oxley Act is changing the way the business world operates. High-quality staff, automation, and processes will be a must-have for all public companies. The long-term payback will ultimately be a significantly higher level of awareness and controls that will produce much higher business processes throughout business units reporting up to their corporate parents.

Sarbanes-Oxley will make the astute programmer, system administrator, and CIO indispensable within their organization. These positions cannot be rationally offshored or outsourced. The personal risk to the people running the corporation is too high. The requirement to attest that the systems are working as intended and described is so intense, only a mad man would send this work halfway around the world to save a couple of bucks. ☺



Jack Martin, editor-in-chief of *WebSphere Journal*, is cofounder and CEO of Simplex Knowledge Company and publisher of *Sarbanes-Oxley Compliance Journal*, the only journal devoted exclusively to Sarbanes-Oxley compliance. Simplex developed the world's first diagnostic-quality ultrasound broadcast system. Jack is co-author of the upcoming book, *Understanding WebSphere*, from Prentice Hall. jack@sys-con.com

A Fresh Brew of Chart FX...



With a Robust Blend of Form and Function.

10 years of creating quality developer charting solutions gives us an edge over others in the industry. So, although we're the new kid on this block, we've been in the neighborhood for a long time. The maturity of Chart FX and our support will prove it. Chart FX for Java is a 100% Java component that uses JSP technology to produce



charts in a variety of formats: PNG, JPEG, SVG and FLASH. Developed using JDK 1.4, it supports J2EE 1.3 and J2SE 1.4. Chart FX for Java is available as a Server-side Bean and an Enterprise Java Bean (EJB) that runs on most popular Java Application Servers, with features like borders, gradients, alpha blending, anti-aliasing and transparency.

Additionally, it will produce native Windows output in the form of a ActiveX or .NET component. Easy data population from JDBC, XML, Text Files, API and other popular data sources. Includes a Designer, which is a stand-alone tool to set the visual attributes of the chart, and a Resource Center containing the Programmers Guide, the Javadoc API and hundreds of samples. *For more information or to download a trial version, visit www.softwarefx.com.*



Chart FX

for JAVA

www.softwarefx.com

In the US: (800) 392-4278 • In the UK: +44 (0) 117 905 8733 • In Germany: 0800-24 27 839

As Java development evolves, so does JProbe®

JProbe®

Find the cause of J2EE code performance, memory and threading problems faster than ever before with Quest JProbe. New investigative features for finding memory problems combined with dramatic performance improvements mean even quicker resolution of problems in your application, servlet, JSP and EJB code.

JProbe Suite

JProbe Profiler

JProbe Memory Debugger

JProbe Threadalyzer

JProbe Coverage

Part of the Quest Performance Management Suite for the J2EE Platform



For more info and a free eval, visit:

<http://www.quest.com/jdj>