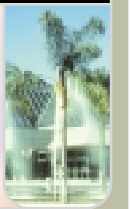
OPEN SOURCE VERSUS **THE JAVA PLATFORM** pg. 6

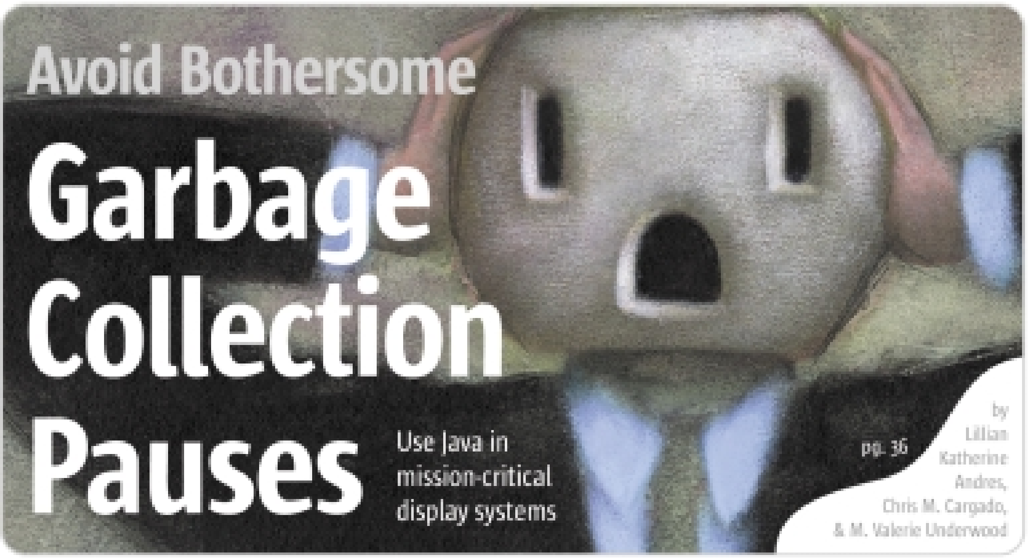
JAVA™ DEVELOPER'S JOURNAL

The World's Leading Java Resource

July 2003 Volume:8 Issue:7

www.JavaDevelopersJournal.com





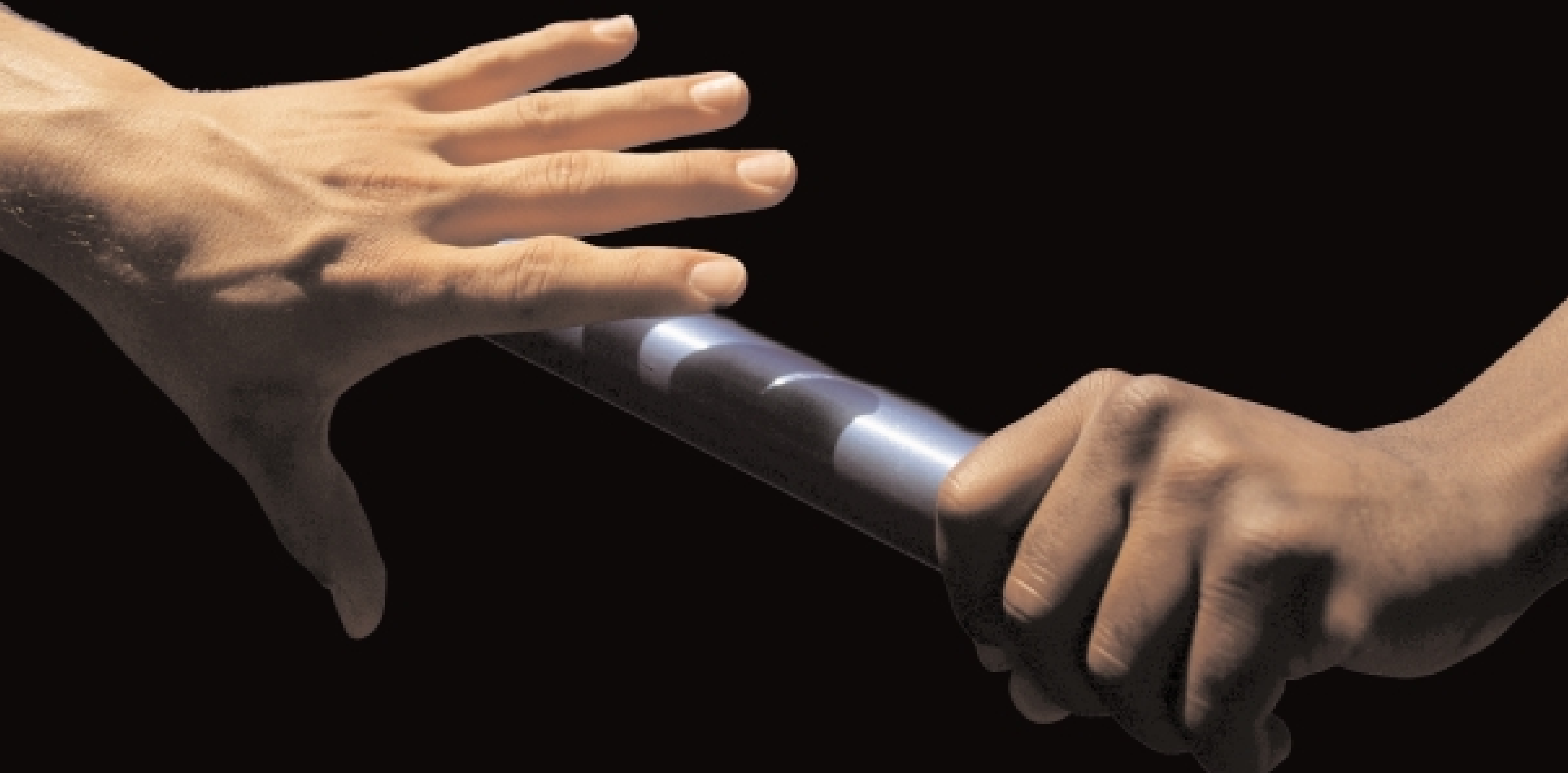
Avoid Bothersome

Garbage Collection Pauses

Use Java in mission-critical display systems

pg. 36

by
Lillian
Katherine
Andres,
Chris M. Cargado,
& M. Valerie Underwood



New JBuilder® 9

WORK THE WAY YOU WANT.
DEVELOP YOUR JAVA™ APPLICATIONS FASTER.
BUILD IN PERFORMANCE AND QUALITY FROM THE START.

Borland® JBuilder® makes the fast faster because it works the way you work. It knows what you need to do next, no matter what your style. JBuilder gives you an active voice at every stage in the development process and is the developer's access point into the Borland Application Lifecycle solution — best-in-class products integrated to help the whole development team make better software, faster. JBuilder — the **#1** development environment for building any type of Java application: Web, Enterprise, Mobile, and Web Services.

Borland®
Excellence Endures™

go.borland.com/j1

DEFINE DESIGN **DEVELOP** TEST DEPLOY MANAGE

Best Laid Plans...

Alan Williamson
Editor-in-Chief

I’ve recently returned from the razzmatazz of the eighth JavaOne in San Francisco. The 2003 conference was characterized by a massive drive back to the developer, with Sun Microsystems attempting to win back our hearts and put its arm around us all in a virtual hug.

Commendable. I fear, however, that Sun may have forgotten who or what the developer is.

Take the keynotes, for example. Usually notable affairs, they set the tone for the coming year and basically cheer us up and embolden us to face the challenges ahead. This year we had three keynotes, all from Sun executives, with Scott McNealy saved for the last day. The conference opened with Jonathan Schwartz, who just doesn’t seem to warm to the geek crowd at all. His keynote on Day One had the lamest of demos that didn’t impress the hard-core Java audience who were waiting (and willing) to be awed.

Sadly, the Rich Green part of Day Two was equally uninteresting. It was at the point when “Project Rave” was unveiled – the tool Sun is counting on to rival Visual Studio for ease of use and speed – that Sun seemed to have forgotten who their audience is. Green was enthusing about how Sun wants to increase the current 3 million Java developers to 10 million, and explained how these 7 million new developers within the Java ecosystem would be called “Java Corporate Developers” – in other words, scripters (or, as I heard one person say, “drag’n’droppers”).

While we all applaud this new move to increase our developer base, it needs to be done in such a way so as not to alienate the current developer community, or patronize or undervalue the existing Java developers in any way while introducing this new breed of developer.

Microsoft, with their legions of Visual Basic developers, has proven that such a community does indeed exist and we really do need to reach out and bring them into the fold somehow. Whether Project Rave is the vehicle to bring them in is yet to be seen. Sun, historically, has not fared too well at creating developer tools, but obviously we’ll reserve judgment until we see it.

Fortunately the level of keynotes picked up immensely the moment James Gosling took the stage. This was pure gold and worth the trip. He was his classic self, drawing out the “geek streak” that’s deep-rooted in us all. Gosling took us through a wide range of different “cool” projects that definitely make you proud to be in the Java field.

Scott McNealy ended the conference with an entertaining and uplifting talk, especially since he’s eased off the Microsoft bashing somewhat and let his own personality come through. Java has finally arrived and I got the impression that he was proud at last to be standing up there stating facts about Java, as opposed to the overhyping of previous years.

Sun also unveiled two new Web sites: www.java.net and www.java.com. The .com site is specifically for consumers, to enable them to come to grips with Java and to get the latest software installed on their machine. The java.net site is aimed at us, i.e., Java (corporate?) developers. It’s a mishmash of SourceForge, JavaBlogs, TheServerSide, and JGuru all rolled into one. It looks good and has the potential to be very popular, even though it’s arguably some five years late in coming!

The question remains, should they have done it at all? Isn’t it a greater validation of our language to have external sites pick up and run with the ball that Sun failed to carry forward so many years ago? Sun will need to navigate its way through the next six months very carefully, very carefully indeed.

Our community demands and deserves respect, and it is up to each and every one of us to make sure Sun doesn’t sell us short. Sun needs to innovate and not emulate (Microsoft) if it is to succeed in growing the developer space for Java.

Alan Williamson, when not answering your e-mails and working on the next issue of *JDJ*, heads up a small team dubbed the “Thunderbirds of the Java industry,” providing on- and off-site rescue for Java projects in trouble. For more information visit www.javaSOS.com. You can also read his blog: http://alan.blog-city.com.

alan@sys-con.com

JAVA DEVELOPER'S JOURNAL

Editorial
Editor-in-Chief: **Alan Williamson**
Editorial Director: **Jeremy Geelan**
Executive Editor: **Nancy Valentine**
J2EE Editor: **Joe Ottinger**
J2ME Editor: **Glen Cordrey**
J2SE Editor: **Jason Bell**
Contributing Editor: **Jason R. Briggs**
Contributing Editor: **Ajit Sagar**
Founding Editor: **Sean Rhody**

Production
Production Consultant: **Jim Morgan**
Associate Art Director: **Louis F. Cuffari**
Associate Editors: **Jamie Matusow**
Gail Schultz
Jean Cassidy
Jennifer Van Winckel
Online Editor: **Lin Goetz**
Technical Editor: **Bahadir Karuv, Ph.d.**

Writers in This Issue
Lillian Katherine Andres, Jason Bell,
Chris M. Cargado, Glen Cordrey, Peter Curran,
Onno Kluyt, Duane Morin, Joseph Ottinger,
Bill Roth, Mark Roth, M. Valerie Underwood,
Xiaozhong Wang, Alan Williamson, Marcus Zarra

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: 201 782-9600

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.

Worldwide Newsstamd Distrubution
Curtis Circulation Company, New Milford, NJ

In Medias Res

Bill Roth

JavaOne always provides plenty of food for thought. JavaOne 2003 was no exception. This year, Alan Williamson, our beloved editor-in-chief, organized a "birds-of-a-feather" session for the ***JDJ*** editorial board. This is quite an auspicious bunch, and this session provided an opportunity for us to meet face-to-face for the first time.

The panel started out a bit slowly, and was initially lightly attended. I suspect this was because it was scheduled early in the evening by JavaOne standards, at 9:30 p.m. But after a few obligatory questions on JDK 1.5 and the JCP, people began streaming in, and the questions became a bit more lively. One of the liveliest topics that came up was about the Java platform and open source.

For the record, while I am still employed at Sun, I am no longer involved in the day-to-day operations of the Java platform. I am still involved in both open source and the Java platform at a personal level. The intersection of these two topics actually hints at a fascinating evolution that is occurring.

The tension between open source and the Java platform is because we are "in medias res." This Latin phrase is used in literature to describe a plot that starts "in the middle of things" (the literal translation). We are in the middle of the plot narrative of nothing less than the evolution of the nature of intellectual property rights.

There are two ways to think about intellectual property (IP) rights. The first is, for lack of a better phrase, the old way. This way says that all intellectual property created must be jealously guarded and carefully licensed, often for commercial gain. This is a time-honored legitimate avenue for exercising your rights. In fact, it was enshrined in Article I, Section 8, of the U.S. Constitution, way back in the 18th century.

There are many (if not all) companies that have taken full advantage of this way of thinking about IP rights, including Sun. These companies have software and other IP jealously guarded and licensed for commercial gain as well as protection. "And that," as Stuart Smalley says, "is okay." In a free society, it must be possible for someone to create something, even if it is only an idea or as ephemeral as software, and be able to profit from it or protect it.

The second way to think about IP rights was born in the late '60s and '70s. At the AI Lab at MIT, a notion arose that more could be accomplished by sharing your IP. Richard Stallman developed this into the CopyLeft and the GNU Public License, which codified this notion. This philosophy asserts that there will be a faster pace of innovation if IP, or code, is shared. It is posited by some that some code is in fact so crucial to the evolution of the art/science that it must be kept "open" forever. There is also proof that points to the success of this approach – OpenOffice.org, Mozilla, and SNORT are key examples.

The conflict between these two views of IP rights will be with us for a long time. It must be possible for you to create something and do what you want with it, free from intervention from the government or a third party. "Do what you want with it" means protect it or make it freely available. What this means, in essence, is that the two ways of thinking about IP rights are essentially irreconcilable. One can not be transubstantiated into the other.

And so this tension around open source will be with us for a long time to come. And this will be what makes JavaOne panels interesting for a long time to come, as well.

Bill Roth is a marketing manager for x86 servers at Sun Microsystems. He previously managed product management and product marketing for J2EE.

br@billroth.net

SYS-CON MEDIA

President and CEO:
Fuat A. Kircaali *fuat@sys-con.com*
Vice President, Business Development:
Grisha Davida *grisha@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 *roybn@sys-con.com*
Director, Sales and Marketing:
Megan Ring *megan@sys-con.com*
Adverting Sales Manager:
Alisa Catalano *alisa@sys-con.com*
Associate Sales Managers:
Carrie Gebert *carrieg@sys-con.com*
Kristin Kuhnle *kristen@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*

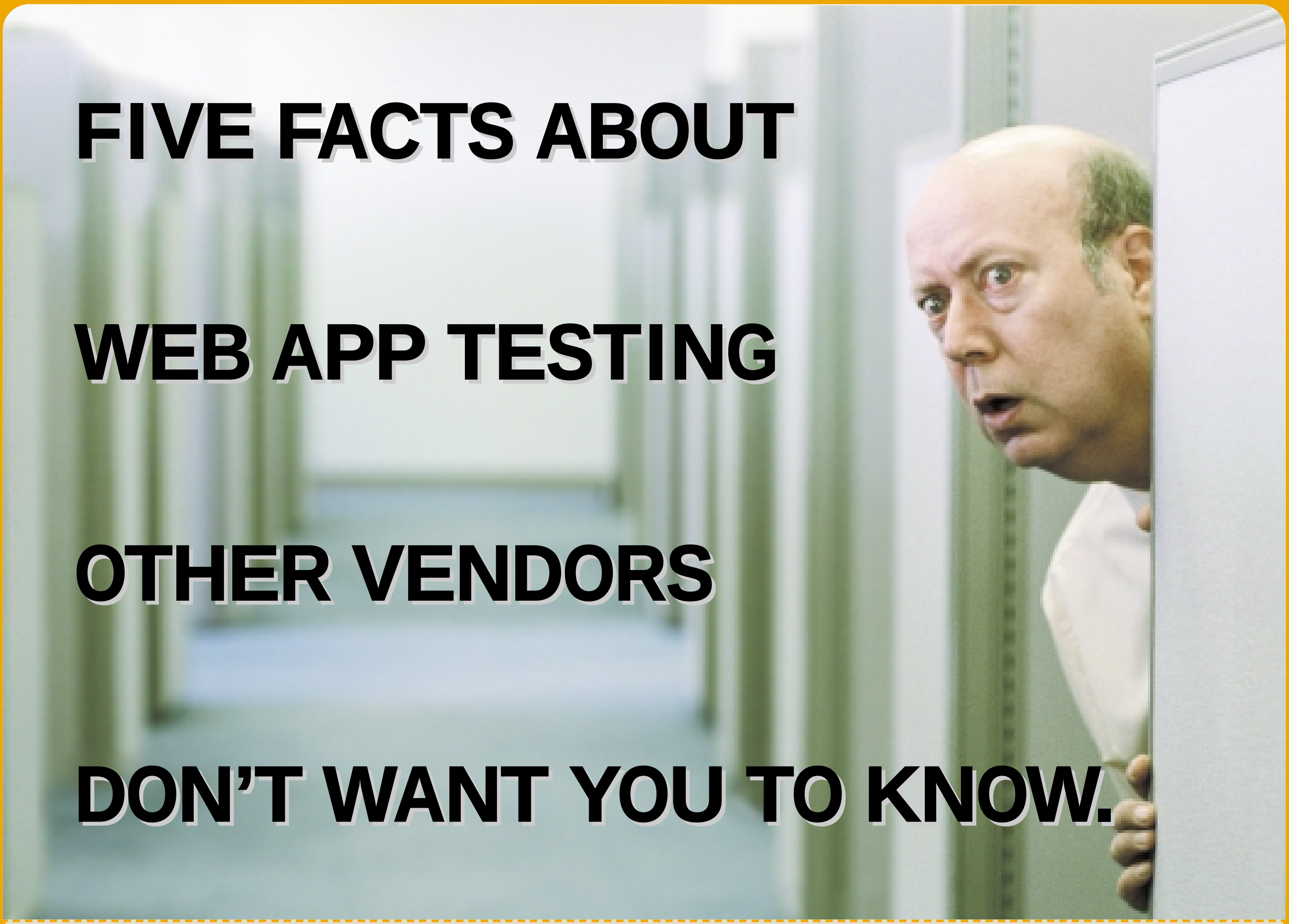
Production
Production Consultant:
Jim Morgan *jim@sys-con.com*
Lead Designer:
Louis F. Cuffari *louis@sys-con.com*
Art Director:
Alex Botero *alex@sys-con.com*
Associate Art Director:
Richard Silverberg *richards@sys-con.com*
Assistant Art Director:
Tami Beatty *tami@sys-con.com*

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:
Kerri Von Achen *kerri@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:
Michael Lynch *michael@sys-con.com*

Customer Relations
Circulation Service Coordinators:
Niki Panagopoulos *niki@sys-con.com*
Shelia Dickerson *shelia@sys-con.com*
JDJ Store Manager:
Rachel McGouran *rachel@sys-con.com*



FIVE FACTS ABOUT WEB APP TESTING OTHER VENDORS DON’T WANT YOU TO KNOW.

With some Web app testing vendors, the thing to ask yourself isn’t "what are they selling" but "what are they hiding"? Consider these five facts:

FACT #1: It’s hard to test dynamic, complex Web apps with a tool originally built to test client-server apps.

FACT #2: It’s even more complicated when you have to manually develop test scripts using a proprietary programming language.

FACT #3: Developing separate scripts for functional tests, load tests, and performance monitoring is inefficient and unnecessary.

FACT #4: You don’t have to put up with restrictive licensing agreements, endless training, and expensive consultants.

FACT #5: You can download our free evaluation and test it against your application in the same time it takes to watch their stale demo.

It’s time to get the facts about Web app testing. See why IDC calls Empirix “a fast-growing leader in Web testing.”* And learn how to get the same results as leading e-TEST™ suite customers like US Bank, British Telecom, and Ceridian.

Get your FREE Web App Testing Fact Pack, including:

- *10 Questions To Ask When Comparing Web App Testing Vendors.*
- Newport Group Whitepaper: *Evaluating Critical Technology Differences Behind Automated Testing Tools*
- Empirix Whitepaper: *25+ Reasons Web Apps Don’t Scale*
- Your own trial copy of e-TEST suite software

To get your Fact Pack:

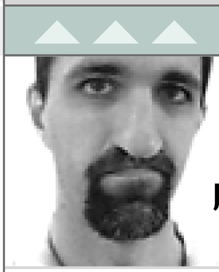
- Call: 1.866.228.3781
- Visit: www.empirix.com/know
- Email: know@empirix.com

empirix

1.866.228.3781
www.empirix.com

Network Computing
Company To Watch 2003

*Source: IDC Bulletin, The Distributed Automated Software Quality Tools Market Forecast and Analysis, 2001-2005, July 2001.



Joseph Ottinger
J2EE Editor

We Need More Innovation

In my last editorial (Vol. 8, issue 6), I argued that we, as an industry, have too much innovation. We have solutions pouring out our ears, stuff we often don't need, yet we use it anyway. This month, I'd like to clarify that somewhat: we need more innovation.

The seeds for innovation are already present: new projects are fertile ground. The problems are often unique, so the solutions that present themselves are individual as well. What's more, sometimes there's a better solution that's simply waiting for the right viewpoint in order to become obvious.

New solutions often imply new technology. Without innovation, we'd be using batch programs to generate paper results overnight. Instead, we had online transactions, the Web, CGI, then mechanisms to improve even that in various ways. All are necessary steps in innovation, and I don't think we've seen the end of online transactions or information presentation yet.

To judge whether we need to create a new solution, we first have to investigate what already exists. We must be willing to accept investigation, especially at a local level, and accept what the results are, even if they go against what we want. We may want to create a new invocation framework; it could be fun to write! However, the costs in terms of development and implementation might not justify the creation of yet another solution.

Once we've accepted that existing solutions aren't going to be enough, it's time to start thinking about how it should be done. This is where the creative juices start flowing, and new ideas wend their way into the light. This is where old ways of doing things die out in a Darwinian survival of the fittest. We need to be willing to kill bad solutions in favor of better, more flexible ones.

Java, as a community, needs to be prepared to do the same – Java itself is susceptible to being outmoded by new technology. If we want "Java" to survive as a name and brand, we must encourage innovation and accept honest evaluation of its strengths and weaknesses in the market – official approval in the form of the JCP isn't enough to keep Java alive. The technology that the JCP puts out needs adherents, adherents in the real world and not just JSR participants or yea-sayers.

It's hard to understate that last point. The JCP tends to foist new standards on an industry that's often watching the rank and file moving in different directions. Look at JDO: some companies were using fairly popular JDO implementations before the JCP started their JDO specification, and the resulting specification ended up invalidating the prior art, even though the prior implementations didn't need as much repair as the specification might have implied.

Standards are best generated from what people are using, not from what people in a boardroom think should be used. The market moves faster than a standards document can. The open source community understands this. So does Microsoft, who tends to flood the market with new products if only to make sure that they're perceived as innovators. As a result, you see the most impressive things from open source initiatives, which can move faster than Sun seems to want to. Some of these will end up working against Sun's vision of Java, and that's all right.

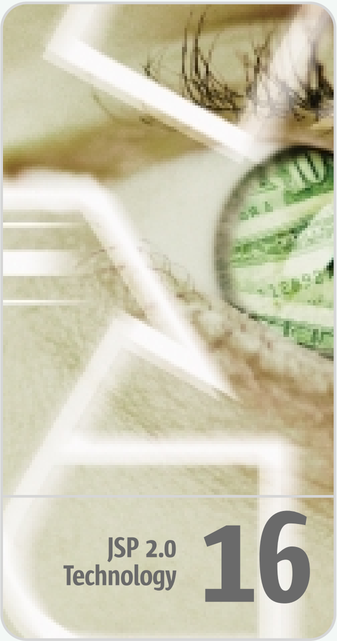
The key is when to innovate and when not to. Innovation should be spurred by fresh, clear ideas about how things might be done better, while acknowledging that industry momentum isn't something to ignore. You shouldn't be creating another solution that duplicates the weaknesses of one that already exists – try to repair the existing one instead.

Therefore, a larger problem is indicated: How do we learn what solutions are out there? There are sites like http://freshmeat.net and others, but those aren't enough; they only echo data that's pushed to them.

If you're working on a project, you should be talking about it, even prematurely. Create an RSS feed, and let various aggregators like http://technews.n-ary.com and http://javablogs.com know about it. Watch these sites and participate in the overall community as much as your time permits. Eventually you learn which way the wind blows, and how to leverage all that information to your benefit.

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.

joeottinger@sys-con.com



JDJ Asks...
IBM-Rational

Interview with Grady Booch

JDJ asked Grady Booch of UML and Rational fame to answer your toughest questions. One of the original developers of the Unified Modeling Language (UML), Booch is recognized internationally for his innovative work in software architecture, modeling, and software engineering processes.

Arun Kumar: *The AOP design method is improving a lot in project development phases. Can you suggest some methods for UML testing?*
Grady Booch: There's a testing profile for the UML underway inside the OMG but, even now, the current practice is toward greater visualization in the testing process. From the outside in, this involves applying use cases and then their associated behavioral diagrams to assert a test case; from the inside out (meaning, in the debugging of a system), XDE has facilities to create sequence diagrams from a running system, thus facilitating tracking down errant behavior, even in a distributed system.

Sanjay Choudhary: *I'm a great fan of yours and have been reading your books since 1996. My question is: Why do we need collaboration diagrams? When should we use them? I always try to convey my point using the sequence diagram. I read a couple of books but none gave a satisfactory answer.*
Booch: I tend to use collaboration diagrams when it's important to indicate the structural relationships among objects; most of the time, I too use sequence diagrams, but when there are structural relationships among the associated objects, it helps to show them in collaboration.

James McGovern: *As a series editor for a prestigious publisher, you are aware of the decline in the technology book marketplace. What should people be reading and publishers publishing?*
Booch: As I walk down the aisles of Borders and Barnes & Noble (and the virtual aisles of Amazon), I notice a preponderance of books whose half-life can be measured in weeks or months, largely because they cover the same ground that associated product documentation addresses, albeit in some cases in a more approachable fashion. Some of these books are interesting and I certainly respect the labor that went into all of them, but there are only a small number of books that really cut at the fundamentals of building quality systems. I'm a voracious reader of both books and journals (and a few Web sites): on the book front, I always appreciate the classics (Knuth, Brooks, anything with Parnas in it); on the journal front I read a variety of professional and trade journals; on the Web front, I'm a regular visitor to Slashdot, ExtremeTech, and SourceForge.

Nenad Nikolic: *What is the current situation in the field(s) of generating executable code from UML? To make this question broader, maybe some scriptable API can be scripted though a model in UML, or there is some UML derivative intermediate language? What does the future look like? Perhaps there's some strong advancement in this regard in a particular field, e.g., information systems or maybe network protocol drivers?*
Booch: Your question is the very essence of model-driven development. Today we know how to generate code from a variety of structural and behavioral models from the UML; the current trend is toward direct executability of models, with the resulting code being, in essence, an assembly language that's invisible to the developer. The low-hanging fruit for this space lies in business rules, deployment, and schemas, where products exist that already do so. As for an intermediate representation, a lot of what's gone inside UML 2.0 tightens the underlying semantics and metamodel, and you'll see that bear fruit in tooling soon. EMF (the Eclipse Modeling Framework: http://dev.eclipse.org/viewcvs/indextools.cgi/~checkout~/emf-home/main.html) is another good example of the trajectory toward richer underlying metamodels that play with the UML.

George Phillips: *What do you think about Extreme Programming compared to the more traditional design approach that you favor? Do you think there's room for both in a good developer's toolkit?*
Booch: I'm a founding board member of the AgileAlliance (www.agilealliance.org/home); I'm a strong believer in agile methods. In fact, at Rational, we've worked with Robert Martin's company to create an agile plug-in for the RUP. While I still have concerns about scale, what strikes me about agile methods in general is that they address many of the social dynamics among team members that amplify good engineering practice – test-driven development, pair programming, and the creation of a stream of executable releases are all really sound ideas. Kent Beck and I have publicly debated the issue, and I'd conclude that the focus on architecture is the one element that separates my world view from XP: in my experience, a focus on architecture first helps to drive out technical risk, but only when followed by a regular stream of executable releases; in Kent's world view, architecture is something that emerges.

Dennis Ceglenski: *How does UML development process map to the traditional development process of data modeling, activities, and the interaction of the data and activities?*
Booch: Let me start not with the process but with the architecture. In our world view, the architecture of a system involves many views that are woven together. One of these views (the logical view) addresses, among other things, the vocabulary of the problem space, and that in turn involves traditional schemas; another view (the component view) addresses the packaging of these things, such as into databases; the deployment view addresses the distribution of these physical databases

Application Server Experts Agree

"Enterprises that are seeking a well-appointed midtier solution will definitely want to give Oracle9*i*AS a test drive."

InfoWorld

"Oracle9*i*AS is now a formidable application server."

The Middleware company

"Oracle: one stop app server shop."

eWEEK
THE ENTERPRISE NEWSWEEKLY

"[With Oracle9*i*AS,] Oracle is delivering what enterprises need to build a robust, scalable ebusiness infrastructure."

IDC
Analyze the Future

Download a free trial now at oracle.com/download.

ORACLE®

Sources:
InfoWorld, November 2002
The Middleware Company, September 2002
eWeek, September 2002
IDC, May 2002

oracle.com/experts
or call 1.800.633.1072

“There’s some interesting work going on **in the modeling of Web services with the UML”**

through the system. From the process perspective in the RUP, you attack each of these views in parallel, driven by the highest risk. In some domains, it’s possible to defer these traditional data decisions; in most others, it’s an early and important part of the evolution and construction of a system.

Adeeb Khan: *A software design validation tool – like running a Unit test in which the results are either success or failure – is there any such tool through which we can test a design for its scalability, concurrency control, security, and other architectural features? If there’s not, what about the possibility of developing one, at least to validate some aspects of software architecture.*

Booch: These nonfunctional requirements are typically attached to functional use cases, and so through these use cases you test these elements. As for performance, I suggest you look at the work by Lloyd Williams on modeling performance with the UML.

Bahram: *In USDP what is the best time to fix a data model with the end user? What is the maximum iteration that we can have in medium-scale projects?*

Booch: I have to plead ignorance as to what USDP is, sorry. As I indicated earlier, though, there’s no “best time”…it really depends on the business and technical context, although if you are risk driven with validation through a stream of executable releases, you’ll rarely go wrong. I have to offer a similar answer to your second question: it really depends on context, although I’d advise you to run your process with a regular rhythm of releases, each mapped to a set of use cases and each representing a successive refinement of the system’s architecture.

Sajan Thomas: *I’m not a good OO programmer and have more experience with DBA, data modeling, and conventional programming. My experience with UML is in using use cases and some UML modeling tools to generate diagrams. One thing I noted in the UML approach (I’m not sure of the validity of this question) is that it differs in relational data modeling in the representation of the one-to-many relationship. Is there any reason? Also, will the philosophy of UML grow to a level that helps us bring relational data, OO data, and OO programming under one umbrella? I don’t see a clear way of getting it done using UML.*

Booch: Take a look at the UML User Guide and you’ll see some examples of traditional database modeling with the UML; also, you ought to read Eric Naiburg’s book *UML for Database Design* on the same topic.

David E. Gonzalez: *What is your opinion on the future of UML, Java, XML, or any other technology. Do you see the possibility of some kind of “grand unifying” approach to these flexible, portable, and open technologies? In addition, where can I find some real-world examples of UML applications for teaching purposes?*

Booch: Well, Java’s not the last language, although it may be Scott McNealy’s last language…

The UML has very much entered the mainstream of development; model-driven development will drive the UML even deeper into various parts of the development process. My personal opinion is that we won’t see a purely executable UML, but rather it will coexist with traditional textual languages. In the longer term, I’m keeping my eye on aspect-oriented development, which addresses issues of costcutting concerns.

As for teaching resources, Rational has a university program (SEED), www.rational.com/corpinfo/college_relations/seed/index.jsp, and you may find what you’re looking for there.

Joe: *What are your thoughts on the usefulness and realities of solutions that claim to automatically take you all the way from a UML model to a generated complete Web application solution? Are they real or pipe dreams?*

Booch: For certain well-defined domains and platforms these things are a reality (note Rational’s architected rapid application development tools). The general solution, however, is hard, although the UML 2.0 semantics go a long way to help make this possible for broader domains.

Joe: *Rational’s UML Resources Site hosts the 1999 PDF of UML Web Modeling extensions, and there’s Jim Conallen’s recently updated book on the subject, but that appears to be about all I can find on Web Modeling Extensions (the 1999 PDF says to look at the Rational UML site for updates, but none can be found). Are the Web Modeling Extensions actually useful for real-world Web modeling? (The book and PDF appear to be geared more toward MVC1 JSP-based designs than to MVC2 controller-based designs). If you guys think this stuff is useful for Web application design, it would be great if you updated it to reflect where J2EE is going with MVC2 designs (e.g., Struts-based apps and so on).*

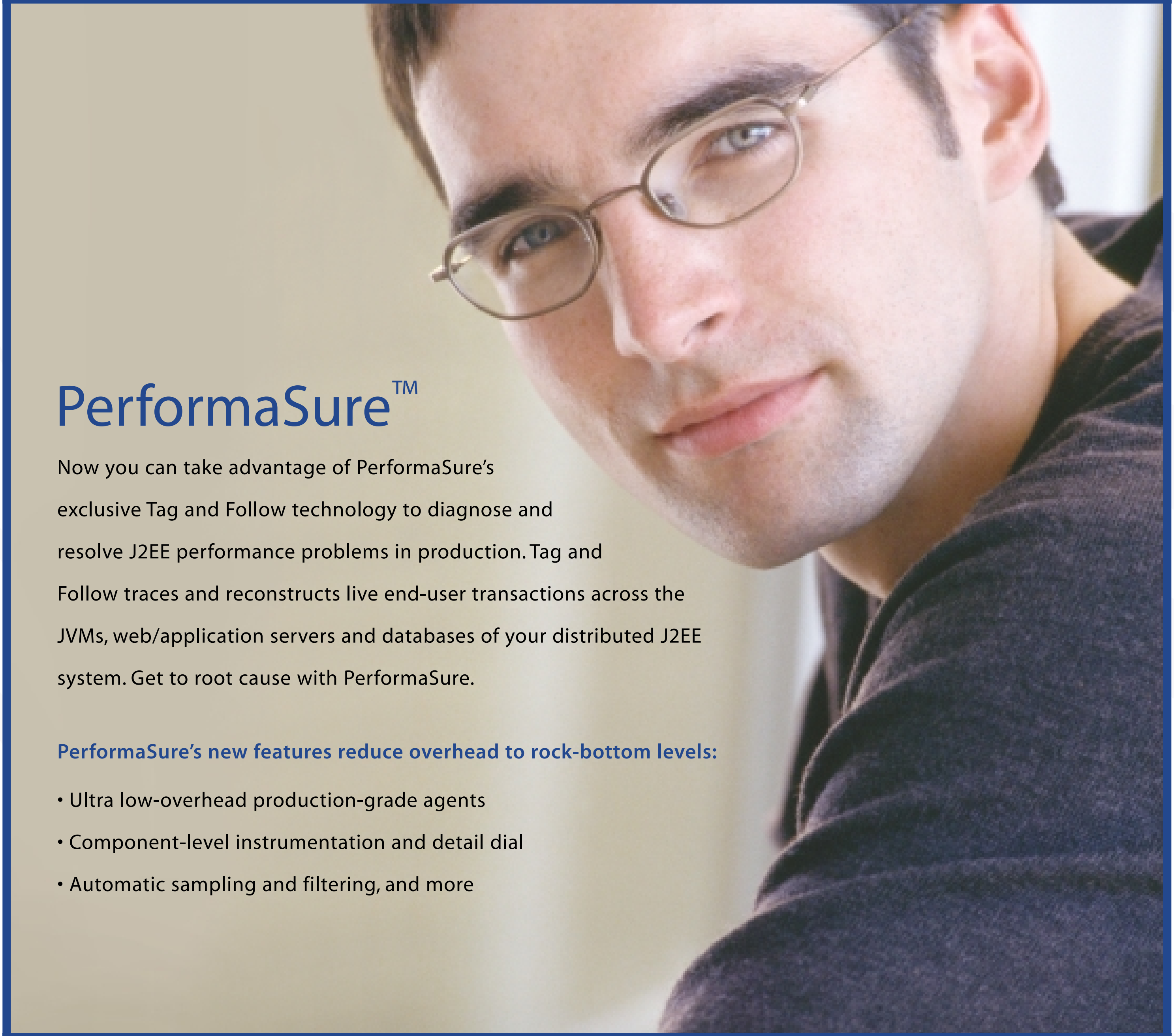
Booch:: I’ll let Jim and our Webmasters know!

There’s some interesting work going on in the modeling of Web services with the UML, by the way.

Brian Wintz: *There seems to be an ongoing battle over development processes – waterfall is viewed as too heavy and inflexible whereas agile/Xtreme/iterative seem to be in vogue. It seems to me that this often heated debate over which is the right approach tends to focus more on personal opinion than on substance. I’m inclined to believe that each development approach has its strengths and weaknesses – speed to market, quality of product, and cost to develop. Could you provide some insight about development processes?*

Booch: The pendulum swings every now and then from high-ceremony to low-ceremony processes. I think the RUP has the right balance: architecture first, followed by the iterative and incremental growth of that architecture through a series of executable releases. In general, it’s a matter of balance: those organizations that tend to favor high-ceremony, waterfallish processes tend to be afraid of confronting risks and producing executable products, and so often hide behind a process; those who tend to favor low-ceremony processes tend to be allergic to control, predictability, and repeatability.

Diagnose and resolve J2EE performance problems …now in production



PerformaSure™

Now you can take advantage of PerformaSure's exclusive Tag and Follow technology to diagnose and resolve J2EE performance problems in production. Tag and Follow traces and reconstructs live end-user transactions across the JVMs, web/application servers and databases of your distributed J2EE system. Get to root cause with PerformaSure.

PerformaSure's new features reduce overhead to rock-bottom levels:

- Ultra low-overhead production-grade agents
- Component-level instrumentation and detail dial
- Automatic sampling and filtering, and more

Part of **Quest Central** for J2EE – **Integrated J2EE application performance management**

QUEST SOFTWARE®

Take a tour of PerformaSure today:

http://java.quest.com/performasure/jdj

by Mark Roth

JSP 2.0 Technology

The community delivers!

JavaServer Pages (JSP) technology originated more than four years ago as a powerful way to dynamically generate HTML on the server side. Over time, and with the input of the developer community, JSP technology has evolved and matured, keeping simplicity at the forefront. The next generation of JSP technology, version 2.0, will be released with J2EE 1.4 and represents an easy-to-use, robust, and extensible technology for building Web applications, well suited toward generating dynamic Web content in such formats as HTML, DHTML, XHTML, SVG, and XML.

The JSP 2.0 specification introduces many new features, including a simple yet flexible integrated expression language, an encapsulation mechanism called tag files, a simplified tag extension API, and a substantially improved XML syntax. Let's explore some of these features and see what they mean for your Web project.

The Evolution of JSP Technology

In its early stages, the focus of JSP technology was to enable the generation of dynamic content by embedding scriptlets (pieces of Java programming language code) within HTML template data. Programming with scriptlets can be quite flexible and powerful at times and is suitable for some types of projects. For many projects, however, such as those that follow a Model/2 or MVC (Model-View-Controller) architecture, embedding scriptlets in the presentation layer can have several disadvantages, including:

1. The Web designers on your project need to learn the Java programming language, which has a fairly steep learning curve in comparison to, say, the JavaScript programming language. Furthermore, pages created with JSP technology ("JSP pages") with scattered pieces of code can become difficult to read and maintain.
2. It becomes too easy to mix business logic with presentation logic, especially when working under a deadline. Even well-intentioned developers working on well-designed Web applications may find it tempting to introduce business logic in the presentation layer.
3. Code becomes more difficult to reuse. Scriptlets frequently lead to copying and pasting of code. Tag extensions are often used to encapsulate and reuse such code. However, until this release of the specification, writing tag extensions has been a tedious and time-consuming process.

JSP technology has evolved in various ways that help make writing pages without inline scriptlets much more of a reality. These evolutions have come in the form of both changes to the specification and add-on technologies. The introduction of tag libraries in version 1.1 of the JSP specification allowed for JSP technology to be extended, and MVC frameworks like Struts began to evolve, providing a simple way to abstract business logic from the presentation layer. After improvements to tag library support in version 1.2 of the JSP specification, the JSP Standard Tag Library (JSTL) was introduced, providing a core set of useful actions such as iteration, internationalization, formatting, SQL database access, and XML manipulation. JSTL also introduced an expression language that's much easier to read and write than scriptlets. Over time, these incremental changes have helped JSP technology become quite suitable for architectures in which it is used purely as a presentation layer, while maintaining strong support for other architectures as well.

The latest JSP specification is currently in the Proposed Final Draft stage and it's under development in the Java Community Process (JCP) as Java Specification Request (JSR) 152. In combination with JSTL, the features introduced in the JSP 2.0 specification yield a cleaner, easier-to-use, and higher-performing language. In fact, page authors using JSP 2.0 technology no longer need to know or use the Java programming language, which in and of itself dramatically decreases their training requirements. Furthermore, the introduction of features like tag files and simple tag extensions enable new reuse patterns and make life easier for tag library developers.

JSP 2.0 technology is expected to have a substantial impact on the way page authors can write JSP pages. Because of this, the expert group decided to upgrade the major version number of the specification. Among other benefits, upgrading the major version number helps differentiate between developing using JSP 1.x technology (with scriptlets) and developing using JSP 2.x technology (with simple expressions and JSTL). It's important to note that though major version number upgrades often connote a break in backward compatibility, this is not the case here. Version 2.0 of the JSP specification is fully backward compatible with version 1.2.

Simple Expression Language

The JSTL 1.0 specification included a simple expression language, intended to help reduce the amount of scriptlets in a page, and to make it much easier for a page author to access application data from within a JSP page. JSTL's expression language was originally called SPEL (Simplest Possible Expression Language), and is intentionally very similar in syntax to ECMAScript (JavaScript) and XPath. By popular demand from the community, this simple expression language is now built into the JSP 2.0 specification.

Reports!

EMBED REPORTS IN JSP, SERVLETS, & JAVA APPLICATIONS.
DEPLOY ON YOUR EXISTING J2EE APPLICATION SERVER.

Invoices. Statements. Activity summaries.
Packing slips. Transaction histories. Reports.

USE JAVA OBJECTS AS DATA SOURCES FOR YOUR REPORTS.

Map objects to a relational model, perform queries, and make joins visually with point-and-click wizards.

OUTPUT REPORTS IN ANY FORMAT USERS REQUIRE.

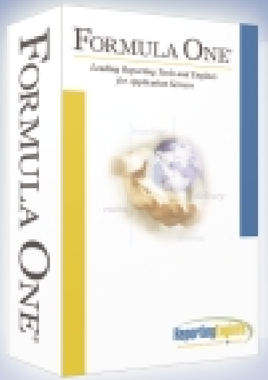
PDF. Excel. XML. DHTML. HTML. Email.

CREATE NEW REPORTS QUICKLY, DYNAMICALLY.

Visual report design environments.
Programmatic creation with APIs.

GET REPORTS UP AND RUNNING QUICKLY.

Includes framework for building a reporting portal on WebSphere, WebLogic, Apache, JBoss, JRun, or any application server.



Formula One e.Report Engine

For PDF, XML, HTML, DHTML and email reports.

Free Trials & Demos: www.ReportingEngines.com/info/JDJ_July_ere.jsp

Formula One e.Spreadsheet Engine

For richly formatted Excel reports.

Free Trials & Demos: www.ReportingEngines.com/info/JDJ_July_ese.jsp

ReportingEngines
JAVA REPORTING TOOLS FROM ACTUATE

888-884-8665 • 913-851-2200
sales@ReportingEngines.com

To understand the motivation for the expression language, let's look at a simple example. Suppose you're a page author writing a JSP page that outputs census data for a given state. The part of your JSP page that outputs the result might look like:

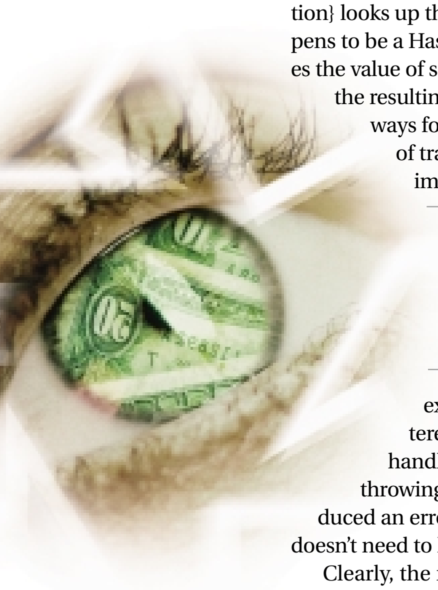
```
The population of <%= state.getFullName() %> in 2000 was
<%
    StateInfo info = (StateInfo)stateInfo.get( state.getId() );
    if( info != null ) {
%>
        <%= info.getPopulation(); %>
<%
    }
%>
```

Writing even a fairly simple example like this requires some knowledge about the Java programming language syntax, types, variable declarations, and the danger of dereferencing a null pointer. This can be intimidating and easy to get wrong for a page author with a limited knowledge of the Java programming language.

Using the simple expression language available in the JSP 2.0 specification, the same presentation logic can be written much more concisely:

```
The population of ${state.fullName} in 2000 was
${stateInfo[state.id].population}.
```

Here's how it works. The expression ${state.fullName} looks up the state attribute, treats it as a JavaBeans component, and calls the accessor for the fullName property (i.e., getFullName()). The expression ${stateInfo[state.id].population} looks up the stateInfo attribute, which in this case happens to be a HashMap; finds the entry with the key that matches the value of state.getId(); and then calls getPopulation() on the resulting entry. Incidentally, the EL behaves in similar ways for maps, lists, and arrays, so there's some degree of transparency in terms of how stateInfo is actually implemented. Another nice feature of the JSTL expression language is if a null value is encountered at any point, the expression language will handle it gracefully and output nothing rather than throwing a NullPointerException that would have produced an error page instead. Of course, the page author doesn't need to know what happens behind the scenes.



"JSP 2.0 technology adds the ability to extend the expression language through writing custom EL functions"

Clearly, the resulting code is more readable and easier to maintain.

Using the JSP 1.2 and JSTL 1.0 technologies, the page author could use these simple expressions only inside the attributes of JSTL actions. JSP 2.0 technology allows page authors to finally use these expressions (now termed "EL expressions") directly within template text, and when passing attribute values to standard actions and tag handlers created with JSP technology ("JSP tag handlers"). That means they can be used with any tag library, even your own, with no extra work required by the tag library developer.

With the power of an integrated expression language, and with the core JSTL actions at hand, a new programming methodology is now available to JSP page authors, making scriptlets a thing of the past and dramatically reducing the learning curve associated with JSP technology. In fact, to help project leads enforce a no-scriptlets policy project-wide or in a specific set of pages, the JSP 2.0 specification now includes a global configuration mechanism to disallow scriptlets for a set of JSP pages in a Web application (among other global configuration options not discussed in this article). This was a popular feature request from project leads who were struggling to prevent developers from mixing business logic with presentation logic, especially under time pressure. Listing 1 illustrates a deployment descriptor configured to disable the use of scriptlets for all JSP pages in the /client subdirectory of the Web application. Of course, for some projects scriptlets will still make sense, and the JSP 2.0 specification supports those types of projects as well.

Writing a New EL Function

Along with a fairly complete set of operators, the expression language in the JSP 2.0 specification has built-in capabilities to access data from maps, lists, arrays of objects, and bean properties. The expression language is designed from the ground up to be intuitive to use and easy to read.

A key decision was made by the expert group to disallow unrestricted invocations on Java methods from within EL expressions. Allowing this could quickly lead to pages that mix business logic with presentation logic, leading to many of the same problems described earlier that existed with scriptlets. As always, the expert group will be observing how people use JSP 2.0 technology to see if it makes sense to loosen this restriction.

To provide a way to add to the power and flexibility of the expression language in a more controlled manner, JSP 2.0 technology adds the ability to extend the expression language through writing custom EL functions. This feature enables a developer to quickly and easily make common tasks available to the page author without having to allow unrestricted access to arbitrary method invocations.

The following code snippet shows a page that invokes an EL function that returns a random number between one and six. The function is imported as part of a tag library (tag libraries can contain both actions and functions) and the function is identified using the prefix of that tag library.

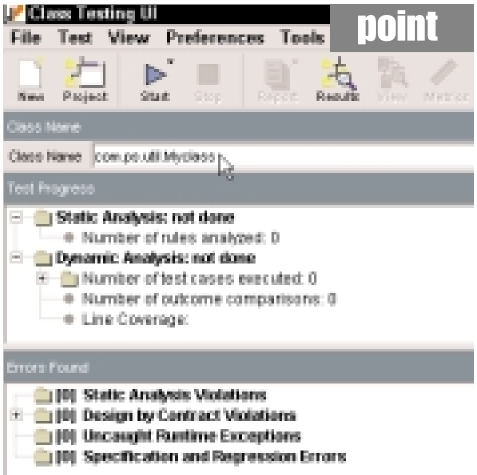
```
<%@ taglib prefix="my" uri="http://acme.com/mytaglib" %>

Congratulations, you rolled a ${my:randomNumber( 1, 6 )}!
```

Writing the implementation of the randomNumber function is quite straightforward. First, the developer writes a public static method:

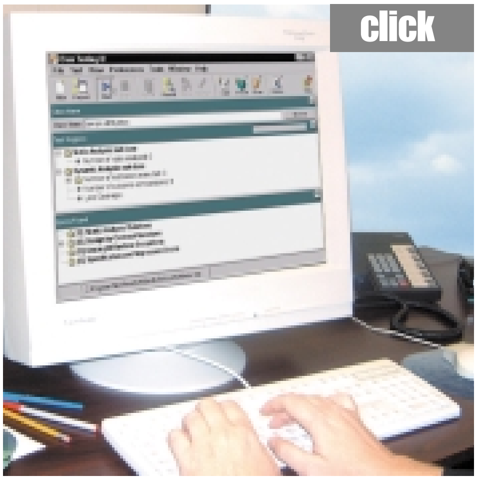
```
/* /WEB-INF/classes/mytaglib/Functions.java */
package mytaglib;

public class Functions {
    public static int randomNumberImpl( int low, int high ) {
        return (int)(Math.random() * (high-low+1) + low);
    }
}
```

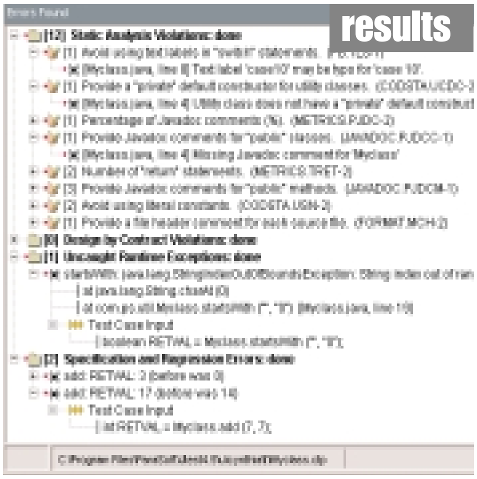
PARASOFT®
"We make software work."

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



```
Class Testing UI
Class Name
Class Name  com.ps.util.Myclass
Test Progress
⊟ Static Analysis: not done
    Number of rules analyzed: 0
⊟ Dynamic Analysis: not done
  ⊞ Number of test cases executed: 0
    Number of outcome comparisons: 0
    Line Coverage:
Errors Found
  [0] Static Analysis Violations
⊞ [0] Design by Contract Violations
  [0] Uncaught Runtime Exceptions
  [0] Specification and Regression Errors
```





```
Errors Found
⊟ [12] Static Analysis Violations: done
  ⊟ [1] Avoid using text labels in "switch" statements. (PB.TLS-1)
      [Myclass.java, line 8] Text label 'case10' may be typo for 'case 10'.
  ⊟ [1] Provide a "private" default constructor for utility classes. (CODSTA.UCDC-2
      [Myclass.java, line 4] Utility class does not have a "private" default construct
  ⊞ [1] Percentage of Javadoc comments (%). (METRICS.PJDC-2)
  ⊟ [1] Provide Javadoc comments for "public" classes. (JAVADOC.PJDCC-1)
      [Myclass.java, line 4] Missing Javadoc comment for 'Myclass'
  ⊞ [2] Number of "return" statements. (METRICS.TRET-2)
  ⊞ [3] Provide Javadoc comments for "public" methods. (JAVADOC.PJDCM-1)
  ⊞ [2] Avoid using literal constants. (CODSTA.USN-2)
  ⊞ [1] Provide a file header comment for each source file. (FORMAT.MCH-2)
⊞ [0] Design by Contract Violations: done
⊟ [1] Uncaught Runtime Exceptions: done
  ⊟ startsWith: java.lang.StringIndexOutOfBoundsException: String index out of ran
      | at java.lang.String.charAt (0)
      | at com.ps.util.Myclass.startsWith ("", "0") [Myclass.java, line 15]
    ⊞ Test Case Input
        | boolean RETVAL = Myclass.startsWith ("", "0");
⊟ [2] Specification and Regression Errors: done
  ⊞ add: RETVAL: 2 (before was 0)
  ⊟ add: RETVAL: 17 (before was 14)
    ⊞ Test Case Input
        | int RETVAL = Myclass.add (7, 7);
C:\Program Files\ParaSoft\Jtest4.5\Jtest\Myclass.ctp
```

(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/jdj2. Or call 888-305-0041.

Special Offer:
For a limited time, every 100th customer to download an eval copy of Jtest automatically wins a fully licensed copy — FREE OF CHARGE.

Platforms:
Linux
Solaris
Windows 2000/XP

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

PARASOFT®
jtest®

The static method can be in any class, and a class can implement more than one EL function.

After the implementation is written, simply add an entry to the tag library descriptor as shown in Listing 2. Now you can use the EL function in any of your JSP pages.

The JSTL 1.1 specification, which will be finalized at the same time as the JSP 2.0 specification, will introduce 16 standardized EL functions, covering common page author needs:

- ***fn:length():*** Get the length of a collection or a string.
- ***fn:toUpperCase(), fn:toLowerCase():*** Change the capitalization of a string.
- ***fn:substring(), fn:substringBefore(), fn:substringAfter():*** Get a subset of a string.
- ***fn:trim():*** Trim whitespace from a string.
- ***fn:replace():*** Replace characters in a string.
- ***fn:indexOf(), fn:startsWith(), fn:endsWith(), fn:contains(), fn:containsIgnoreCase():*** Check if a string contains another string.
- ***fn:split():*** Split a string into an array.
- ***fn:join():*** Join a collection into a string.
- ***fn:escapeXml():*** Escape XML characters in a string.

Incidentally, the tag libraries in JSTL 1.1 have new URIs (for example, http://java.sun.com/jsp/jstl/core instead of the JSTL 1.0 equivalent http://java.sun.com/jstl/core_rt). The new JSTL 1.1 tag libraries accept request-time expressions for their attributes, and delegate to the JSP container to evaluate EL expressions. When creating a new JSP 2.0 application, you should always either (in order of preference, from highest to lowest):

- Use JSTL 1.1
- Use the _rt versions of the JSTL 1.0 tag libraries
- Use the non-rt versions with isELIgnored="true" in the page directive

Tag Files

There are a number of ways to encapsulate reusable portions of JSP software. Since version 1.1 of the JSP specification, there have been four reuse mechanisms: the include directive, the include standard action, the forward standard action, and the custom action.

Of the four, the custom action is the most flexible and powerful reuse mechanism and it has the most readable calling syntax. Custom actions are grouped into tag libraries, which are imported by the page author via a taglib directive. The action is then invoked by inserting a simple XML element. See Listing 3 for an example invocation of a custom action that retrieves products from the user's shopping cart and places them in the "products" variable.

Until version 2.0 of the JSP specification, custom actions could only be implemented using the Java programming language. In version 1.2, the amount of code and configuration required to write custom actions made them typically only worthwhile for complex tasks. The JSP 2.0 specification introduces a new reuse mechanism called a tag file that allows custom actions to be written using JSP technology syntax ("JSP syntax"). This brings the power and flexibility of custom actions to page authors that don't necessarily know the Java programming language.

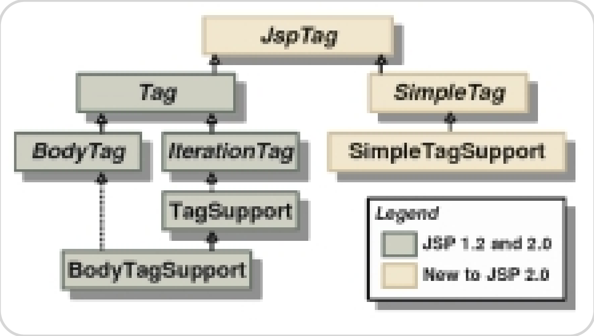


Figure 1 Tag extension class hierarchy

In Listing 3, the <my:queryCart> custom action is used to retrieve a list of products in a shopping cart and place the list in the "products" variable. Some HTML and JSTL code is then used to render the shopping cart in a table. It would be nice to be able to reuse the presentation code that renders the table as well, so that whenever we need to display the user's cart within a JSP page, all we would have to do is type something like <my:showCart />. The cart would be queried and rendered in a single step.

To do this, we could write our own custom action using the Java programming language, but in doing so we would run into some limitations. First, because custom actions are not designed to be invoked from within other custom actions, there would be no convenient way to reuse the existing <my:queryCart> custom action. Second, the purpose of the new <my:showCart> custom action is primarily presentation focused. The implementation would contain many out.println() statements to render the HTML. This is more difficult to read and maintain for the same reasons that servlets that mostly output HTML are more difficult to read and maintain than JSP pages that do the same.

Tag files are the perfect solution to this situation. At its surface, a tag file is simply an easy way to write a custom action. Just as JSP pages are compiled into servlets, tag files are compiled into custom actions. Using a tag file, we can easily construct a custom action that queries the cart and renders the result. Listing 4 is an example of a tag file that does just that.

You'll notice the code for the tag file looks almost identical to the code for the JSP page in Listing 3. One addition is the use of the attribute directive that allows us to specify that this custom action accepts an attribute with the name of "username". The value of the username attribute is then available to the tag using the EL expression ${username}.

The beauty of tag files is that all we need to do now is save this file in /WEB-INF/tags/showCart.tag in our Web application, and we now have a new custom action. Unlike when implementing custom actions using the Java programming language, we don't need to write a tag library descriptor (TLD), and we don't need to manually compile our source code into a tag handler. Also note how easy it is to alter the presentation of the shopping cart versus what it would take to do the same if it were implemented using Java class files.

The following code makes use of our new custom action:

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
Here are the contents of your shopping cart:<br/>
<tags:showCart username="${username}" />
```

From the caller's perspective it's fairly transparent whether the action was implemented using a tag file or a Java class file. The only hint is the use of the tagdir attribute, which can be replaced with uri if we take the extra time to write an explicit TLD file for this tag.

If page authors find themselves frequently copying and pasting a portion of code written in JSP or HTML syntax, perhaps with small changes each time, that portion of code can be placed in a tag file, parameterized, and then reused with minimal effort. Tag files are also a good way to move scriptlets out of your JSP pages and into a more well-defined, encapsulated place.

Of course, not all actions are best implemented as tag files. Tags that are dominated by scriptlets or logic are probably better compiled by hand into Java class files.

Simple Tag Extensions

Writing a JSP 1.2 tag handler using the Java programming language requires a number of tricky steps including choosing from three possible interfaces to implement; implementing the doStartTag(), doInitBody(), doAfterBody(), and doEndTag() methods; picking return values for each method to affect the way the container evaluates the tag body and the rest of the page; and carefully checking the implementation against a complex reuse life cycle. Even some JSP technology experts have trouble getting all these steps right.

Listing 5 shows an implementation of a JSP 1.2 tag handler that repeats the contents of its body a given number of times. Note how what should be a simple loop is spread across two separate methods and how the value of the count attribute is not allowed to be modified by the tag handler implementation so that the tag handler instance may be reused by the container.

JSP 2.0 technology makes writing tag handlers in the Java programming language easier by providing a much simpler tag handler API. The new API is called the simple tag extension API, and JSP 1.2 tag handlers are now referred to as classic tag handlers. Figure 1 illustrates the new tag extension class hierarchy.

When writing a simple tag handler, a tag library developer needs only to extend SimpleTagSupport and implement the doTag() method. Simple tag handlers flow more naturally and are thus easier to write, debug, read, and maintain. The one disadvantage of simple tag handlers is that scriptlets are not allowed in the bodies of these tags. Though some would consider this an advantage, it does mean that simple tag handlers cannot be used for some applications. The tag body is encapsulated in an object called a fragment, which is passed to the tag handler and invoked as many times as needed. Listing 6 shows an implementation of the same repeat action as a simple tag handler.

Tag files are actually translated into simple tag handlers by the container and therefore have many similar properties (for example, scriptlets are not allowed in the body of a tag file invocation either). The following code shows an implementation of the repeat action using a tag file. This is actually the most convenient implementation of the three since it does not require writing a TLD or compiling any classes.

```
/%-- /WEB-INF/tags/repeat.tag --%/

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ attribute name="count" rtexprvalue="true" required="true" %>

<c:forEach begin="1" end="${count}">
    <jsp:doBody />
</c:forEach>
```

Mark Roth is a member of the Java 2 Platform, Enterprise Edition technology team. He has contributed to a number of key specifications and implementations and is currently a co-lead for the JavaServer Pages specification version 2.0.

mark.roth@sun.com

We're Listening!

This was just a quick overview of some of the more significant features new to the JSP 2.0 specification. There are many other features that were not touched on in this article, including:

- ***Greatly improved XML syntax:*** Provides a natural fit for pages that are provided in XML syntax or that generate XML content.
- ***Configuration:*** Allows central control over various properties of JSP pages from within the deployment descriptor.
- ***Portable debugging support through JSR-45:*** Enables the freedom to mix and match IDEs and application servers from different vendors, for those that support this standard.
- ***Dynamic attributes:*** Enables tag extensions to process an open-ended set of attributes.
- ***Enhanced I18N support:*** Allows specification of page encoding on a per-file basis, among other long-awaited enhancements.
- ***Fragment attributes:*** Effectively allows a single tag extension to accept multiple tag bodies.

The new features in the JSP 2.0 specification, when used with other Java technologies like the JSTL tag library, bring power with simplicity to all users of JSP technology, from the basic page author who no longer needs to learn the Java programming language to the advanced tag library developer who can now write powerful tag handlers with much less overhead.

The expert group created the new features in the JSP 2.0 specification directly from input received from the Java technology developer community at large – from people like you. Download the JSP 2.0 specification and an implementation like J2EE 1.4 SDK Beta 2 (which is based on Tomcat 5) and try these features out for yourself. Aside from support for the JSP 2.0 and Servlet 2.4 specifications, Tomcat 5 has several enhancements over Tomcat 4.0 including improved performance and a rewritten code generator that overcomes the 64K method size limitation, among other things.

Your feedback has already made, and continues to make, a big difference!

Resources

Implementations

- *J2EE 1.4 SDK Beta 2:* http://java.sun.com/j2ee/1.4/download-beta2.html
- *Java Web Services Developer Pack:* http://java.sun.com/webservices/
- *Tomcat 5:* http://java.sun.com/products/jsp/tomcat/
- *Jakarta Taglibs project:* http://jakarta.apache.org/taglibs/
- *TLDDoc (JSP Tag Library Documentation Generator):* http://developers.sun.com/dev/coolstuff/tlddoc/

Information

- *JSP 2.0 tutorial:* http://java.sun.com/j2ee/1.4/docs/tutorial/
- *JSP 2.0 specification:* http://jcp.org/en/jsr/detail?id=152
- *JSP:* http://java.sun.com/products/jsp/
- *JSTL:* http://java.sun.com/products/jsp/jstl/
- *JavaServer Faces Technology:* http://java.sun.com/j2ee/javaserverfaces/

Community

- *java.net Web Applications:* http://java.net/

Acknowledgments

Special thanks to Eduardo Pelegri-Llopart, Pierre Delisle, Stephanie Bodoff, and Jan Luehe, whose feedback helped me improve this article.

InstallShield MultiPlatform

Mac OS X

Solaris

Linux

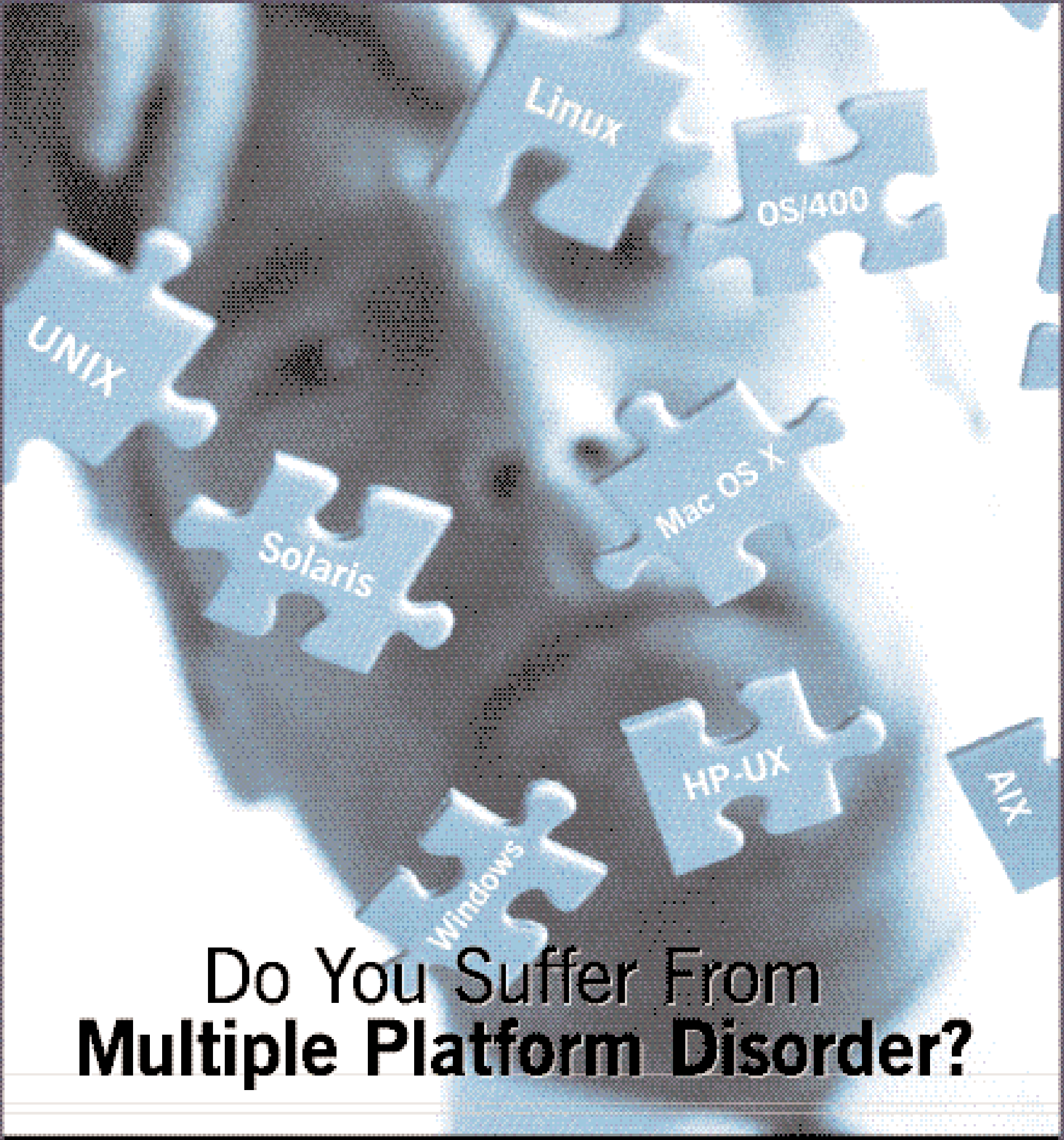
AIX

HP-UX

OS/400

UNIX

Windows



Do You Suffer From **Multiple Platform Disorder?**

>> Get InstallShield MultiPlatform – the ultimate cure major OS vendors trust.

So your application needs to install on multiple operating systems, but you're suffering from wasted development time, broken installers, higher project costs, and frustrated customers. Sounds like you've got a bad case of Multiple Platform Disorder. You better get InstallShield MultiPlatform, the MPD cure trusted by major software vendors like IBM, HP, Sun, PeopleSoft, Siebel, SAS, and J.D. Edwards. With MultiPlatform, you can easily build a single installer that deploys onto virtually any platform, including Mac OS X, OS/400, Windows, AIX, Solaris, Linux, HP-UX, or any other flavor of UNIX.

- Unparalleled Flexibility
- Total Platform Support
- Open API
- Extensive Java Support
- Powerful Wizards
- High-Productivity IDE

Say goodbye to Multiple Platform Disorder for good!

Download a free evaluation copy of InstallShield MultiPlatform 5.0 today.

www.installshield.com/puzzlejdj

IBM Business Partner

InstallShield.

Listing 1 Using JSP configuration to disable scriptlets

```
<!-- /WEB-INF/web.xml -->

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-app_2_4.xsd"
           version="2.4">
     <jsp-config>
          <jsp-property-group>
               <url-pattern>/client/*.jsp</url-pattern>
               <scripting-invalid>true</scripting-invalid>
          </jsp-property-group>
     </jsp-config>
</web-app>
```

Listing 2 Defining a new EL function

```
<!-- /WEB-INF/tlds/mytaglib.tld -->
...
<function>
   <description>
      Returns a random number in the provided range.
   </description>
   <name>randomNumber</name>
  <function-class>mytaglib.Functions</function-class>
  <function-signature>
      int randomNumber( int, int )
   </function-signature>
   <example>
      my:randomNumber( 1, 10 )
   </example>
</function>
...
```

Listing 3 JSP Page to display a shopping cart

```
 <%@ taglib prefix="my" uri="http://acme.com/mytaglib" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

Here are the contents of your shopping cart:<br/>

<my:queryCart var="products" user="${username}" />

<table>
   <c:forEach var="product" items="${products}">
      <tr>
        <td>${product.name}</td>
        <td>${product.price}</td>
      </tr>
   </c:forEach>
</table>
```

Listing 4 Tag files encapsulate reusable JSP code

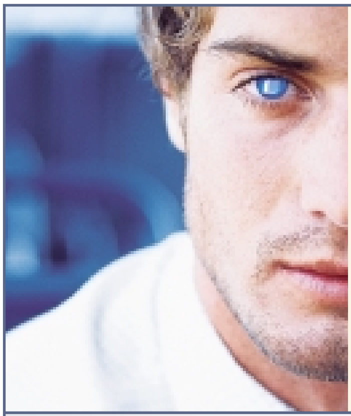
```
<%-- /WEB-INF/tags/showCart.tag --%>
<%@ attribute name="username" rtexprvalue="true" %>
<%@ taglib prefix="my" uri="http://acme.com/mytaglib" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core %>

<my:queryCart var="products" user="${username}" />

<table>
   <c:forEach var="product" items="${products}">
      <tr>
        <td>${product.name}</td>
        <td>${product.price}</td>
      </tr>
   </c:forEach>
</table>
```

Listing 5 Repeat Action implemented as a “classic tag handler”

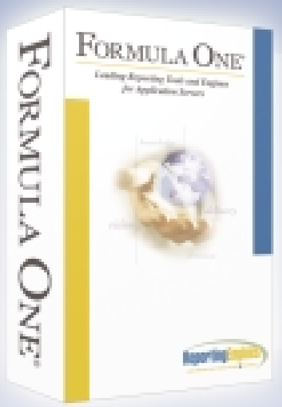
```
 /* /WEB-INF/classes/mytaglib/RepeatClassicTag.java */
package mytaglib;

import javax.servlet.jsp.tagext.BodyTagSupport;

public class RepeatClassicTag
      extends BodyTagSupport
{
      private int count;
      private int timesRemaining;

      public int doStartTag() {
           this.timesRemaining = count;
           return EVAL_BODY_INCLUDE;
      }

      public int doAfterBody() {
           int result = EVAL_BODY_AGAIN;
           this.timesRemaining--;

           if( this.timesRemaining == 0 ) {
                 result = SKIP_BODY;
           }

           return result;
      }

      public void setCount( int count ) {
           this.count = count;
      }

      public int getCount() {
           return this.count;
      }
}
```

Listing 6 Repeat Action implemented as a “simple tag handler”

```
 /* /WEB-INF/classes/mytaglib/RepeatSimpleTag.java */
package mytaglib;

import java.io.IOException;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.SimpleTagSupport;

public class RepeatSimpleTag
      extends SimpleTagSupport
{
      private int count;

      public void doTag()
           throws IOException, JspException
      {
           for( int i = 0; i < count; i++ ) {
                 getJspBody().invoke( null );
           }
      }

      public void setCount( int count ) {
           this.count = count;
      }

      public int getCount() {
           return this.count;
      }
}
```



Reports!


FORMULA ONE
FORMULA ONE
Leading Reporting Tool and Engines for Application Servers
ReportingEngines

J2ME
J2SE
J2EE
Home

Jason Bell
J2SE Editor

Sleeping Tigers

J2SE is going through a bit of an overhaul at the moment, with the release of J2SDK 1.5 (project name "Tiger") due at the end of 2003. Sun Microsystems ran a feature article in May about this release that included a Q&A with Joshua Bloch, a senior staff engineer at Sun (http://java.sun.com/features/2003/05/bloch_qa.html). 1.5 contains enhancements that enable developers to create simple robust code. The impact on legacy code has been kept to a minimum. Well, I hope so; at least that's the way I read it.

I use Generics on a daily basis and find it an interesting issue, especially when dealing with RSS/RDF data items (and trust me I see a lot of RSS data).

- Old style:

```
List myList = new ArrayList();
```

- New style:

```
List myList<String> = new ArrayList<String>();
```

Now is that funky or what? Do you see the benefits? First, it solves most of those annoying casting problems; no longer do we mere mortals have to suffer typing:

```
Iterator i = myList.iterator();
while(i.hasNext(){
  String thing = (String)myList.next();
}
```

Now we can save ourselves from typing eight characters. Which, from my point of view, is 80 characters a day, multiply that by a year (I work hard), that's 29,200 typing actions saved. Depending on your half-empty/half-full perspectives, you could either be keeping RSI at bay or losing on productivity.

With Generics in 1.5 the errors are caught at compile time, not at runtime. In theory most of the ClassCastExceptions will be eliminated. Sounds like a good deal to me. The jury is still out on whether anyone will go back and change every casting from a collection, but that remains to be seen.

I won't go into the enhanced for loop, autoboxing, or avoiding the creation of boilerplate coding with metadata. The best thing you can do is read Sun's J2SE 1.5 article, while I readjust my cynicism chip to full.

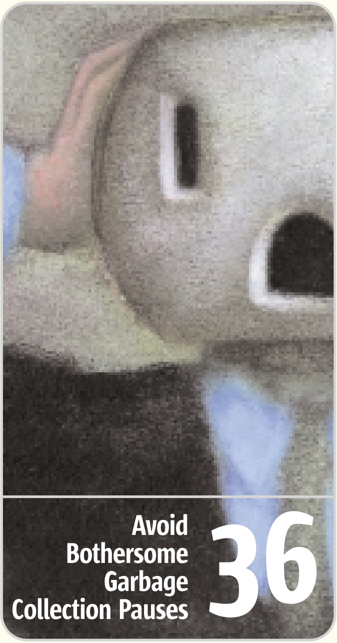
Here we go: Will the 1.5 SDK be available across all OS types? Now I use Windows, Debian Linux, and FreeBSD in a commercial setting so I need a stable SDK across all of these. As for the rest of the world, there's still all the other Linux releases, Solaris (I think Sun will have that one covered), and Mac OS X. I'm always wary of new SDK releases as they take time to be accepted, and then you have to think about the developer catch up. Is everyone now using NIO and Regular Expressions these days? I don't think so, judging by the amount of queries I see in mailing lists and other developer/student help channels.

Developer releases are great – they evolve, invigorate, and charge the head, the hands, and the heart to code like you've never coded before. The question you have to ask is: "Is this release really going to impact the business that I work in?" How is it going to increase productivity? How much retraining will be required? How many more books will I have to buy on the Java language? The ultimate question is: What is the cost benefit?

J2SE 1.5 could be a sleeping Tiger for a good 12 months unless the hearts and minds of Java programmers are refreshed in the things they do. I'm excited about it, but other developers don't always think the same way I do. Sun needs to work hard at serving the community across operating systems. Now I can appreciate that there are always manpower constraints, but a proper roll out has to be maintained. I want to use 1.5 across the OSs I use, but may not be able to do so for a little while yet.

Jason Bell is the senior programmer for a B2B portal. He's also a keen supporter of people reading the API docs before asking questions. In his spare time he's involved with building RSS development tools.

jasonbell@sys-con.com



TAKE THE LEAD
▪ WEBSPHERE TRAINING
▪ PROJECT MENTORING
▪ COURSE LEASING

Trimming the Fat from Swing

Improve performance

Marcus S. Zarra

I'm sure we've all heard it before: Java on the client is slow; Swing is slow. The reality is that Sun has made great progress in increasing the speed of Swing and Java on the client.

However, it's up to developers to demonstrate that Java has indeed improved to the point of usability and viability on the client.

To do this, the code needs to be very lean and clean. One of the original problems with GUI creation was the reliance upon Visual Development Tools to design and code the GUI. While these VDTs have come a long way in the last few years, I have found that many developers rely so heavily on the VDTs that they've gotten lazy when it comes to coding the actual functionality of the application.

The Problem

Methods are expensive in terms of execution time. The more method calls made, the slower the application will be. However, there needs to be a balance between speed of execution and good coding standards.

Swing makes a large number of method calls during its startup and execution. This is a price that has to be paid for its design model. What compounds the method calls that Swing requires is the number of method calls the average developer piles on top of Swing.

Multithreading Swing

The Swing API is considered to be single threaded and thread "unsafe." However, for a Swing GUI to be responsive, some actions must be threaded, such as database access.

In a normal GUI construction, if a call to a database is made, the GUI will just sit there waiting for the database access to return. Not only will the GUI be unresponsive, it won't even repaint! This can cause the user to believe the application has locked up or failed in some way. This problem is not unique to Java. Every GUI API has to deal with this same issue in some fashion. It's unfortunate that many Java Swing developers don't handle this type of situation properly.

Steve Wilson of Sun Microsystems is quoted as saying that in any situation where it's known that the process is going to take a long time, a GUI should respond to the user in some fashion within 50 milliseconds. Any slower than this and it will feel sluggish. Naturally a database request is not going to return in 50 milliseconds or less! This is where multithreading comes into play. The problem lies in properly threading your Swing GUI application to avoid complications with Swing's single-threaded nature.

The solution to this problem lies within two methods: invokeLater and invokeAndWait. These two methods were originally in the SwingUtilities class but, as of v1.2, they've been moved to the java.awt.EventQueue class. Applications can still call these methods in the SwingUtilities class but are merely wrappers for the java.awt methods.

Listing 1 demonstrates database access being spun off into a separate thread. (Listings 1–4 can be downloaded from www.sys-con.com/java/sourcec.cfm.) The method that's called is invokeAndWait, which means the worker thread is blocked until the database activity returns. However, the GUI's thread is not blocked and will allow repaints, and more. *Note:* This is a very primitive example designed to only show the multithreading.

Once the database call has returned, it's time to update the GUI. Since this is still in a separate thread, it won't modify the GUI directly. Thus, in Listing 1, all the GUI updating is in a call to the invokeLater method. This method doesn't block the worker thread that allows it to terminate peacefully. The work to update the GUI has been placed into the event queue, and once it reaches the top of the heap, the GUI will be updated. This allows the update to the GUI to happen from a thread outside the event thread without causing problems.

Model, View, Control

Example 1: JTable

The cleanest way to solve this problem is to write Swing code by hand as it provides two clear benefits:

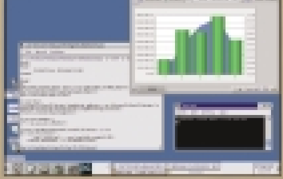
1. The code will be very lean and clean.
2. The code will be much easier to maintain since there will be fewer method calls in the code.

For example, consider a simple JFrame with a few buttons and a table on it. Any complex GUI will generally be using GridBagLayout as its layout manager, therefore this example will be using GBL as the layout manager. Hopefully, through this example, you'll see a reduction in the lines of code, enabling you to write cleaner, tighter, and better performing GUI code.

Listing 2 shows an example GUI built using JBuilder's VDT. I'll admit I was very impressed at how compact the code was that JBuilder produced. There's very little improvement that can be done at this point. However, looking at Listing 3, you can see that we were able to remove a few method calls. The calls to setText() have been moved into the constructor calls for the buttons. The other change we can see at this point is excess construction of GridBagConstraints objects. This is excessive due to what the GridBagLayout does with these objects.

When the GridBagConstraints is passed into the GridBagLayout, the first thing that's done is it's cloned. The GridBagLayout does this so that the GridBagConstraints doesn't change on it unexpectedly. Thus, the individual creation of GridBagConstraints here is unnecessary. This is why, in Listing 4, there's only one GridBagConstraints and it's modified for each call to the JPanel's add method.

Both of these changes are relatively minor and won't improve the performance all that noticeably. However, this is where the VDT stops and the devel-



oper takes over.

It's very common for a developer to utilize the shortcuts in the creation of a JTable to populate it for data. While these shorten the development cycle slightly, they'll cost a fortune in performance. Instead, the creation of an actual table model that's customized for a specific use will improve performance. By implementing your own TableModel, you can reduce the number of method calls and object creations that are normally incurred when using the default models.

For example, take a JTable created with the (Vector, Vector) constructor. The only way you can access that data programmatically is via those vectors. However, the GUI doesn't know if anything has changed inside those vectors and therefore needs to be informed of the change so the GUI can repaint itself. However, if a custom TableModel object is created that's specifically designed to handle the data you wish to represent, you're able to add methods directly to the model that will allow you to edit the data and then fire off an event that the View will intercept and handle. This eliminates a large number of method calls and indirection in your code and indirectly increases the performance of your GUI overall.

Listing 4 is an example TableModel implementing the various methods that are required. Notice that the example extends from the AbstractTableModel. There are a few nonrequired methods in the AbstractTableModel that are useful and don't need to be replaced. This allows the AbstractTableModel to handle the listeners and deal with delivering the events that you create. This leaves us with the methods shown in Table 1.

Our example is fairly simple but it gives you an idea of how you could construct a TableModel specifically tailored to the data it will be representing. It's a rare situation indeed where you will need to display completely dynamic data.

Example 2: JTree

Another area of performance contention is the JTree API. In a large number of cases where I've seen the JTree used in a production environment, it's invariably used incorrectly, causing the number of objects involved with it to double.

The JTree API is designed around the concept of nodes; each point in the tree is either a branch or a leaf. Every point in the JTree implements the TreeNode interface. This interface defines methods inside it that tell the View whether or not that particular node is a leaf or a branch.

What happens is that developers tend to use (read overuse) the DefaultMutableTreeNode class instead of developing their own TreeNodes. This produces a "wrapper" class around

getColumnClass(int columnIndex)	Every object in a particular column needs to be of the same class. In our example, we merely grab the first row and return the getClass() of the object in this column.
getColumnCount()	In the example, we know how many columns of data we're going to have so we have the column count hard coded to avoid unnecessary method calls.
getColumnName(int columnIndex)	Again we know how many columns we have so we have the column names hard coded in a string array and merely return the correct index.
getRowCount()	Our data is all contained within one Vector inside the model, so we merely have to return the size of the Vector.
getValueAt(int rowIndex, int columnIndex)	We know which data objects we'll be containing as well as which pieces of data are going into which columns. Therefore once we have retrieved the correct row of data (element in the Vector), we fall into a switch statement to call the correct getter method on the object.
isCellEditable(int rowIndex, int columnIndex)	Returns a boolean value letting the GUI know whether or not this cell can be edited by the user. This doesn't stop you from altering the data programmatically.
setValueAt(Object aValue, int rowIndex, int columnIndex)	This method does not need to be overloaded. If your table does not allow editing to be done, you can skip this method.

Table 1

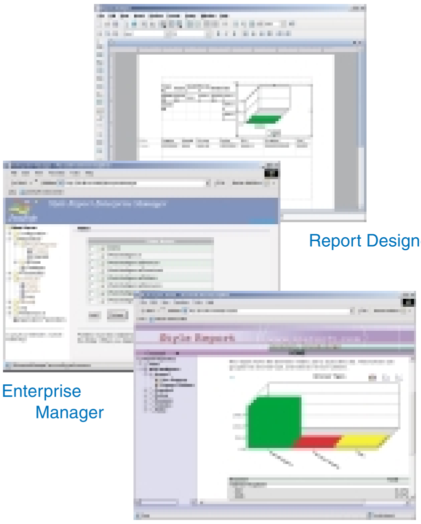
children()	This method returns an enumeration of the children that this node has. If this node is a leaf, it returns null.
getAllowsChildren()	Returns a boolean whether this node can have children. Note that it doesn't necessarily have to have children at this time.
getChildAt(int childIndex)	Returns the child at the specified index.
getChildCount()	Returns how many children this node has.
getIndex(TreeNode node)	Returns the index of the referenced child.
getParent()	Returns the parent of this node. Each node will need to keep a reference to its parent. This could cause a slight reconstruction in the way that you initially construct your tree.
isLeaf()	Simple boolean method that returns whether or not this node is a leaf. Note that a leaf is a node that cannot have children, not one that doesn't currently have any children.

Table 2

Java & Web Reporting:
with Java flexibility
without Java complexity

Style Report:
Information Vantage Point

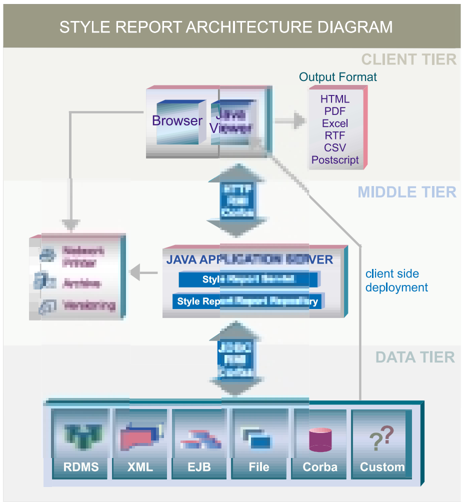
Style Report blends the best of Java and Web to provide an award winning Web enterprise reporting solution. Based 100% on open standards such as XML, JSP, Java and Web services, Style Report allows you fully leverage existing IT infrastructure and common technical skills. One Report, Multiple Channel 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!



Report Designer

Enterprise Manager

Web Report Viewer/Composer



STYLE REPORT ARCHITECTURE DIAGRAM

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

InetSoft
open standards innovation

www.inetsoft.com
JDJ special http://www.inetsoft.com/jdj

Phone: 888-216-2353 (USA)
Phone: 732-424-0400 (Intl)

> “The Swing API is more complicated **than other graphical APIs on the market”**

Marcus S. Zarra is the director of technology at Extraquest Corporation in Denver, Colorado. He began developing professionally in 1985 for school systems and then for legal firms in Phoenix, Arizona. Marcus has been a Java developer since 1996, working in the banking, telecommunications, and insurance industries.

mzarra@extraquest.com

their actual data that causes a doubling of the number of classes in the JTree. While this certainly works (even the tutorials from Sun do this), it's not the most efficient way to handle a JTree.

To remove this doubling of objects in the tree, one of two things can be done:

1. Don't use any data objects inside the DefaultMutableTreeNode.
2. Have your data objects that are going to be represented by a tree implement TreeNode.

Clearly, not putting any of your data objects in the tree is self-defeating, so we'll explore the second option – implementing the TreeNode interface. We don't need to implement the actual model since the default model expects and knows how to handle the TreeNode interface. Thus our data objects that will be displayed in the Tree need to implement the methods shown in Table 2.

None of these methods impact the data you're representing and they have very little to do with the GUI. Thus your design integrity stays intact. However, this allows you to modify the data directly, notify the model that the data has changed, and not have to go through an intermediary layer and deal with casting from Object, and so on. This eliminates numerous method calls and the creation of objects that serve very little purpose.

Conclusion

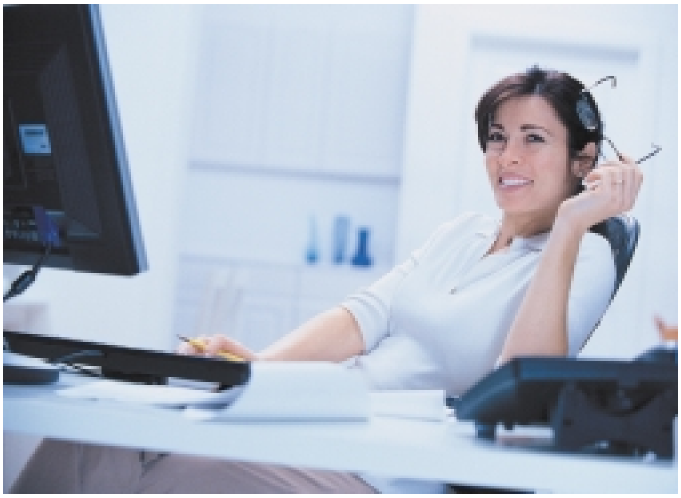
The Swing API is more complicated than other graphical APIs on the market. A large portion of this is attributed to the Model-View-Control design pattern. However, as developers we can avoid compounding the complexity of Swing by reducing the amount of code that we lay on top of it.

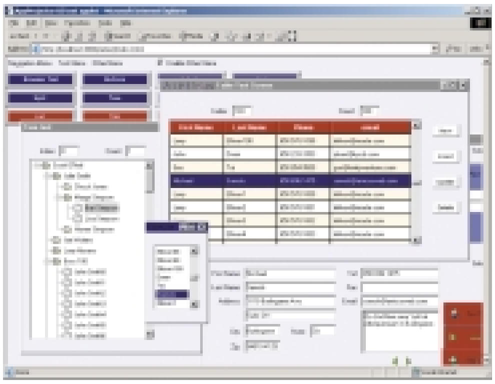
When you go through the optimization phase of your project, look at each method dealing with the GUI and see if there are any method calls that you can remove, any objects that are unnecessary, and any objects you can avoid creating by implementing the model interfaces yourself instead of creating wrapper objects. Once that's complete, start looking for operations that take a long time and consider putting them into separate worker threads instead of the primary GUI thread.

These simple steps will dramatically improve the performance of Swing while still adhering to the OO paradigm as well as the single-threaded rules.

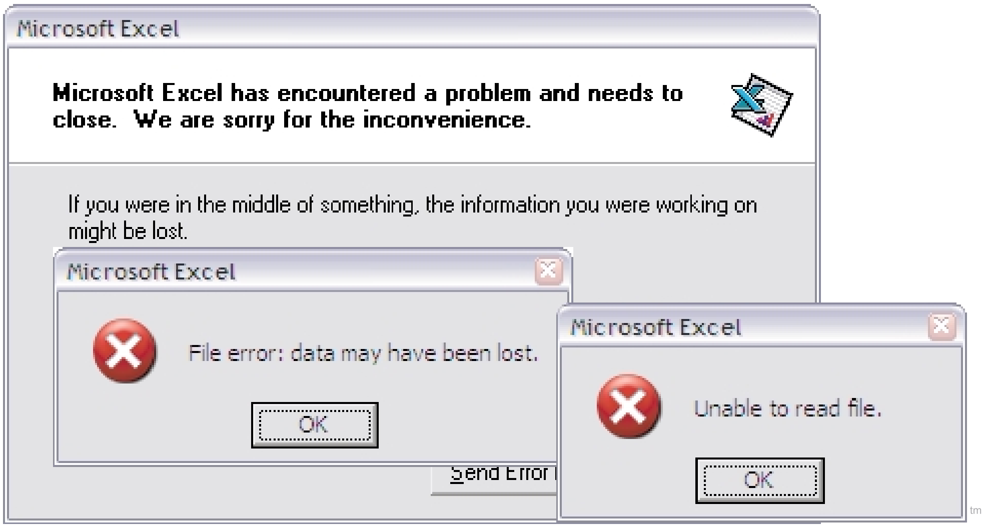
References

- *The Swing Connection:* http://java.sun.com/products/jfc/tsc/
- *Java Platform Performance: Strategies and Tactics:* http://developer.java.sun.com/developer/Books/performance/
- *Christmas Tree Applications:* http://java.sun.com/products/jfc/tsc/articles/ChristmasTree/





is it really free...



Microsoft Excel

Microsoft Excel has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Microsoft Excel

File error: data may have been lost.

Microsoft Excel

Unable to read file.

if it costs you a customer?

Other Java™ Excel™ tools claim... they "have the best support reading and writing Excel™ workbooks currently available." So we modified and wrote some of our customer's Excel™ files using both our competitor's tools and ExtenXLS with identical code.*

The results? While ExtenXLS reliably generated error-free Excel™ files, our competitor's Excel™ output was either corrupted or opened with errors – and that was when it didn't crash on reading the file!

Best-in-class Excel™ compatibility, robust features, and a company dedicated to its products.
That's why companies such as HP, Ford, JP Morgan, GE, and Disney trust ExtenXLS – and you can too.
ExtenXLS – the better Java™- Excel™ API.

Java is a Registered Trademark of Sun Microsystems Inc. Excel is a Registered Trademark of Microsoft Corporation. Jakarta POI is a Registered Trademark of the Apache Software Foundation.
*Comparison based on ExtenXLS.jar version 2.2 dated 5/15/03 and Apache POI version 1.10 (jakarta-poi-1.10.0-dev-20030222.jar). Copyright 2003 Extentech Inc.

JDJ Sale: Get 10% off ExtenXLS Java™|XLS Toolkit through July 2003 * *

Use this special link to get your discount: http://www.extentech.com/jdjsale/

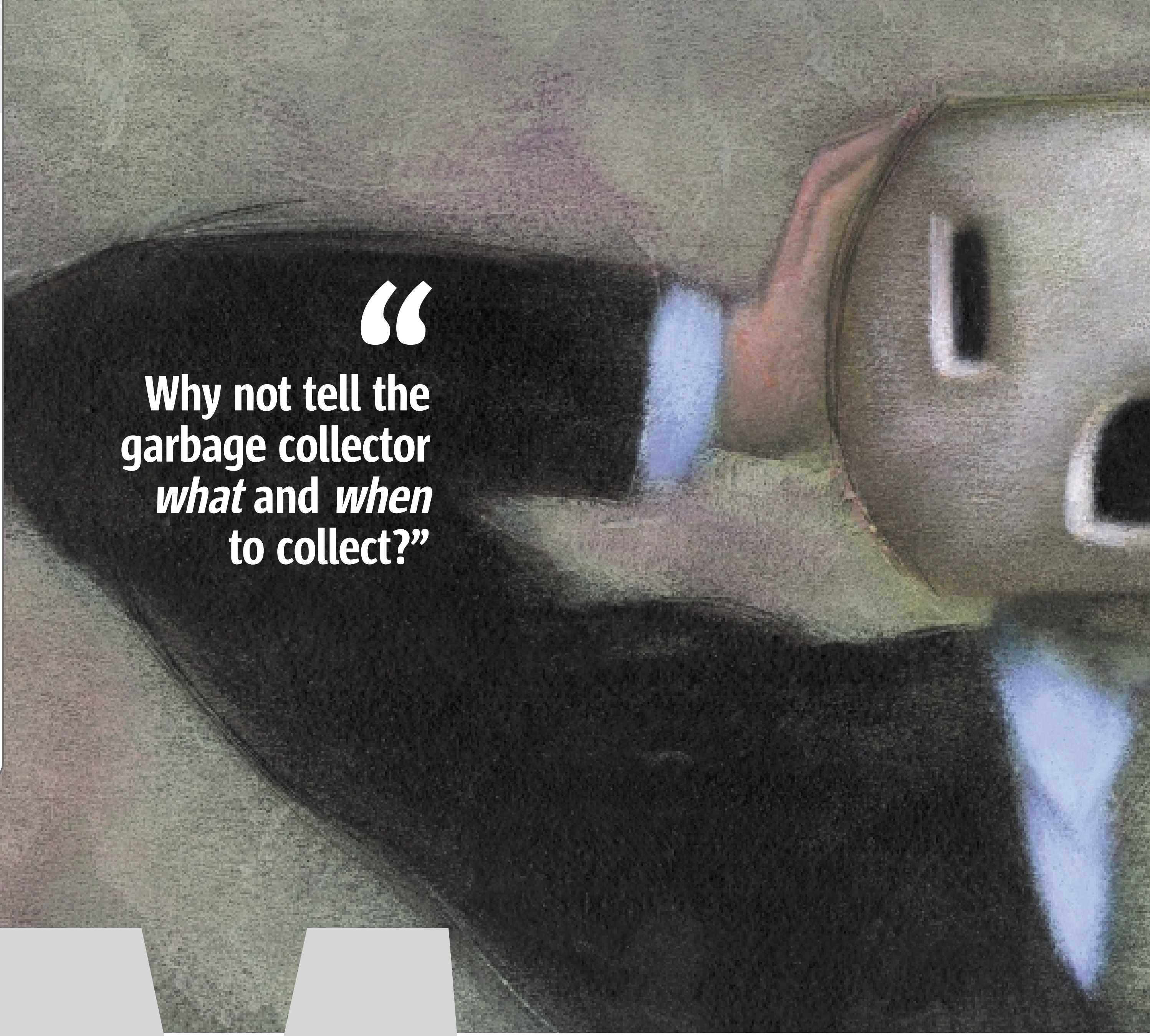
* * 10% off ExtenXLS Java|XLS Toolkit full version and ESD version only. Not valid with other offers.

extenXLS version 2

- ✓ Fully supported
- ✓ Commercial-grade Excel™ compatibility
- ✓ Move and re-order WorkSheets
- ✓ Protect and unprotect WorkSheets
- ✓ Best-in-class 3D Named Range support
- ✓ Add hyperlinks to Cells for drill-down reports
- ✓ Serialize and copy Sheets between WorkBooks
- ✓ Total control of fonts and cell formatting
- ✓ Comprehensive sample code and templates
- ✓ Streamlined, easy to use API
- ✓ Add Java™ Dates directly to WorkSheets
- ✓ Used worldwide by global 1000 companies

extentech

sales@extentech.com



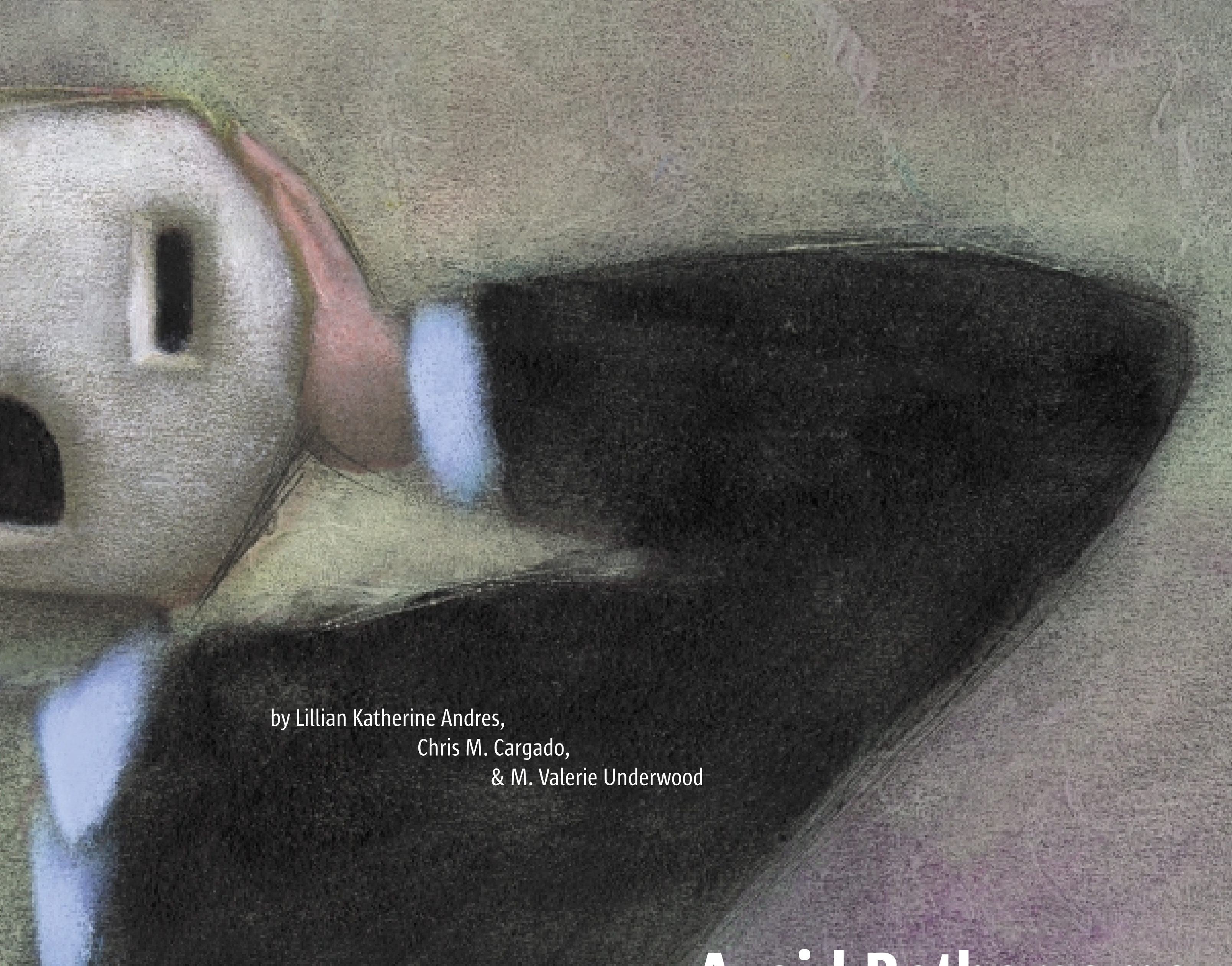
> “Why not tell the garbage collector *what* and *when* to collect?”

Many engineers complain that the non-deterministic behavior of the garbage collector prevents them from utilizing the Java environment for mission-critical applications, especially distributed message-driven displays (GUIs) where user responsiveness is critical. We agree that garbage collection does occur at the worst times: for example, when a user clicks a mouse or a new message enters the system requiring immediate processing. These events must be handled without the delay of in-progress garbage collection. How do we prevent these garbage collection pauses that interfere with the responsiveness of an application (“bothersome pauses”)?

We have discovered a very effective technique to prevent bothersome garbage collection pauses and build responsive Java applications. This technique or pattern is especially effective for a distributive message-driven display system with soft real-time constraints. This article details this pattern in three simple steps and provides evidence of the effectiveness of the technique.

Pattern to Control Garbage Collection Pauses

The Java environment provides so many benefits to the software community – platform independence, industry momentum, a plethora of resources (online tutorials, code, interest groups, etc.), object-oriented utilities and interfaces (collections, network I/O, Swing display, etc.) that can be



by Lillian Katherine Andres,
Chris M. Cargado,
& M. Valerie Underwood

Avoid Bothersome Garbage Collection Pauses

Use Java in mission-critical display systems

plugged in and out – that once you have experienced working with Java it's hard to go back to traditional languages. Unfortunately, in some mission-critical applications, like message-driven GUIs that must be very responsive to user events, the requirements force you to take that step backward. There's no room for multiple second garbage collection pauses. (The garbage collector collects all the "unreachable" references in an application so the space consumed by them can be reused. It's a low-priority thread that usually only takes priority over other threads when the VM is running out of memory.) Do we really have to lose all the benefits of Java? First, let's consider the requirements.

A system engineer should consider imposing requirements for garbage collection like the following list taken from a telecom industry example (see References).

1. GC sequential overhead on a system may not be more than 10% to ensure scalability and optimal use of system resources for maximum throughput.
2. Any single GC pause during the entire application run may be no more than 200ms to meet the latency requirements as set by the protocol between the client and the server, and to ensure good response times by the server.

Armed with these requirements, the system engineer has defined the worst-case behavior in a manner that can be tested.

The next question is: How do we meet these requirements? Alka Gupta and Michael Doyle make excellent suggestions in their article (see References). Their approach is to tune the parameters on the Java Virtual Machine (JVM). We take a slightly different approach that leaves the use of parameter definitions as defined by the JVM to be used as a final tuning technique.

Why not tell the garbage collector *what* and *when* to collect?

In other words, control garbage collection via the software architecture. Make the job of the garbage collector easy! This technique can be described as a multiple step pattern. The first step of the pattern is described below as "Nullify Objects." The second step involves forcing garbage collection to occur as delineated in "Forcing Garbage Collection." The final step involves either placing persistent data out of the reach of the collector or into a data pool so that an application will continue to perform well in the long run.

Step 1: Nullify Objects

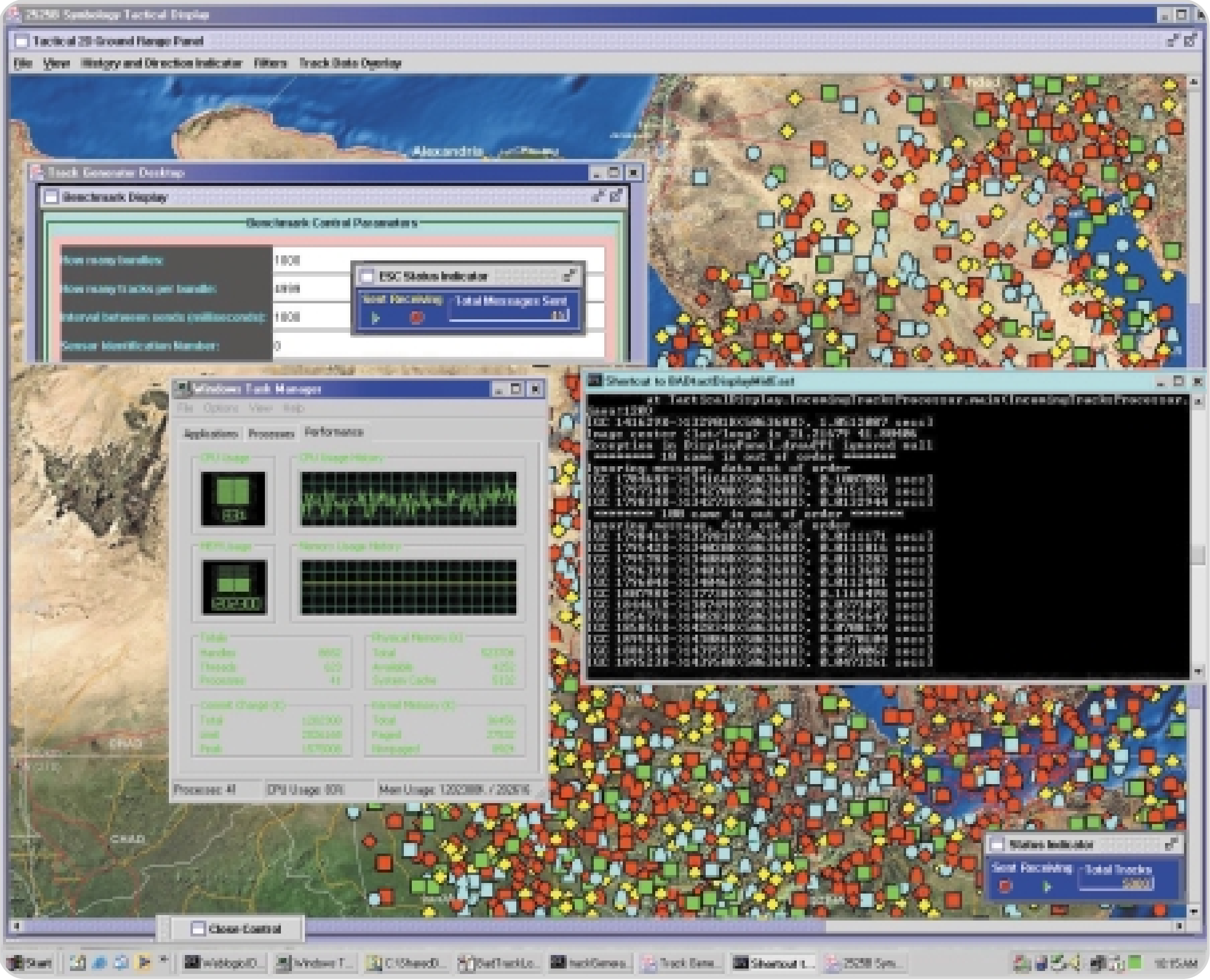
Memory leaks strike fear into the hearts of programmers! Not only do they degrade performance, they eventually terminate the application. Yet memory leaks prove very subtle and difficult to debug. The JVM performs garbage collection in the background, freeing the coder from such details, but traps still exist. The biggest danger is placing an object into a collection and forgetting to remove it. The memory used by that object will never be reclaimed.

A programmer can prevent this type of memory leak by setting the object reference and all underlying object references ("deep" objects) to null when the object is no longer needed. Setting an object reference to "null" tells the garbage collector that at least this one reference to the object is no longer needed. Once all references to an object are cleared, the garbage collector is free to reclaim that space. Giving the collector such "hints" makes its job easier and faster. Moreover, a smaller memory footprint also makes an application run faster.

Knowing when to set an object reference to null requires a complete understanding of the problem space. For instance, if the remote receiver allocates the memory space for a message, the rest of the application must know when to release the space back for reuse. Study the domain. Once an object or "subobject" is no longer needed, tell the garbage collector.

Thus, the first step of the pattern is to set objects to null once you're sure they're no longer needed. We call this step "nullify" and include it in the definition of the classes of frequently used objects.

The following code snippet shows a method that "nullifies" a track object. The class members that consist of primitives only (contain no additional class objects) are set to null directly, as in lines 3–5. The class members that contain class objects provide their own nullify method as in line 9.



How many bundles: 1000
How many tracks per bundle: 4999
Interval between sends (milliseconds): 1000
Sensor Identification Number: 0

```
        at TacticalDisplay.IncomingTracksProcessor.main(IncomingTracksProcessor.
[illegible]
Image center (lat/long) is 21.21679 41.88486
Exception in DisplayPanel.drawPP ignored null
 ======== 10 came in out of order ========
ignoring message, data out of order
[illegible]
 ======== 100 came in out of order ========
ignoring message, data out of order
[illegible]
```

Figure 1 Tactical display

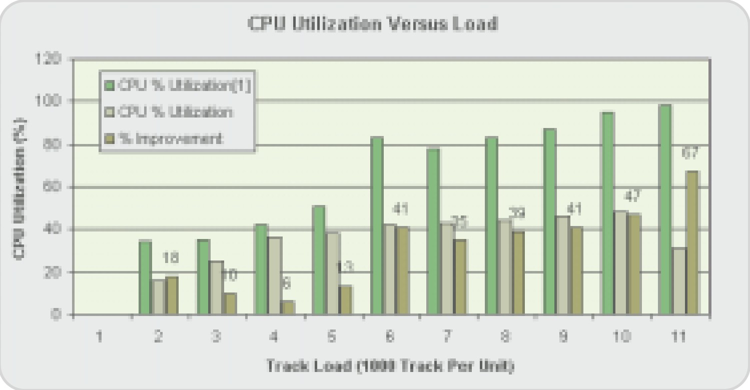


Figure 2 Performance bar chart

```
public void nullify () {

this.threadId       = null ;
this.elPosition     = null ;
this.kinematics     = null ;

  if (this.iff != null)
{
        this.iff.nullify();
        this.iff = null ;
  }
}
```

The track nullify is called from the thread that has completed processing the message. In other words, once the message has been stored or processed, that thread tells the JVM it no longer needs that object. Also, if the object was placed in some Collection (like an ArrayList), it's removed from the Collection and set to null.

By setting objects to null in this manner, the garbage collector and thus the JVM can run more efficiently. Train yourself to program with "nullify" methods and their invocation in mind.

Step 2: "Force" Garbage Collection

The second step of the pattern is to control when garbage collection occurs. The garbage collector, GC, runs as Java priority 1 (the lowest priority). The virtual machine, VM, runs at Java priority 10 (the highest priority). Most books recommend against the usage of Java priority 1 and 10 for assigning priorities to Java applications. In most cases, the GC runs during idle times, generally when the VM is waiting for user input or when the VM has run out of memory. In the latter case, the GC interrupts high-priority processing in the application.

> **"Typically, most engineers want proof before changing their approach to designing and coding"**

Some programmers like to use the "-Xincgc" directive on the Java command line. This tells the JVM to perform garbage collection in increments when it desires. Again, the timing of the garbage collection may be inopportune. Instead, we suggest that the garbage collector perform a full garbage collection as soon as it can in either or both of two ways:

1. ***Request garbage collection to happen as soon as possible:*** This method proves useful when the programmer knows he or she has a "break" to garbage collect. For example, after a large image is loaded into memory and scaled, the memory footprint is large. Forcing a garbage collection to occur at that point is wise. Another good area may be after a large message has been processed in the application and is no longer needed.
2. ***Schedule garbage collection to occur at a fixed rate:*** This method is optimal when the programmer does not have a specific moment when he knows his application can stop shortly and garbage collect. Normally, most applications are written in this manner.

Listing 1 introduces a class named "BetterControlOfGC". It's a utility class that provides the methods described earlier. There are two public methods: "suggestGCNow()" and "scheduleRegularGC(milliseconds)" that respectively correspond to the steps described earlier. Line 7 suggests to the VM to garbage collect the unreachable objects as soon as possible. The documentation makes it clear that the garbage collection may not occur instantaneously, but experience has shown that it will be performed as soon as the VM is able to accomplish the task. Invoking the method on line 25 causes garbage collection to occur at a fixed rate as determined by the parameter to the method.

In scheduling the GC to occur at a fixed rate, a garbage collection stimulator task, GCStimulatorTask, is utilized. The code extends the "java.util.timer" thread in line 10. No new thread is created; the processing runs on the single timer thread available beginning with the Java 1.3 environment. Similarly, to keep the processing lean, the GC stimulator follows the Singleton pattern as shown by lines 18–23 and line 27. There can be only one stimulator per application, where an application is any code running on an instance of the JVM.

We suggest that you set the interval at which the garbage collector runs from a Java property file. Thus you can tune the application without having to recompile the code. Write some simple code to read a property file that's either a parameter on the command line or a resource bundle in the class path. Place the command parameter "-verbose:gc" on your executable command line and measure the time it takes to garbage collect. Tune this number until you achieve the results you want. If the budget allows, experiment with other virtual machines and/or hardware.

Step 3: Store Persistent Objects into Persistent Data Areas or Store Long-Lived Objects in Pools

Using persistent data areas is purely optional. It supports the underlying premise of this article. In order to bind the disruption of the garbage collector in your application, make its job easy. If you know that an object or collection of objects would live for the duration of your application, let the collector know. It would be nice if the Java environment provided some sort of flag that could be placed on objects upon their creation to tell the garbage collector "–keep out". However, there is currently no such means. (The Real-Time

WebAppCabaret™

J2EE Web Hosting

www.webappcabaret.com

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.

***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 and for $25 off the total price of your order please log on to **http://www.webappcabaret.com/jdj.jsp**

Guaranteed Latest Tomcat/JBoss/Jetty for New Enterprise Accounts

Features	Basic	Professional	Enterprise	Dedicated
Monthly Cost	us$9.00	us$17.00	us$39.00	us$191.00
Servlet Spec	2.3	2.3	Latest	Latest
JSP Spec	1.2	1.2	Latest	Latest
EJB Spec	1.1	1.1	Latest	Latest
Private JVM	x	x	x	x
Database	1	1	5	Unlimited
e-Commerce		x	x	x
Tier 1 Center	x	x	x	x
Accounts/Server	120	120	35	1
RAM/Server	2GB	2GB	2GB	256MB
RAID Diskspace	30MB	200MB	900MB	40GB
Bandwidth/Month	3GB	10GB	15GB	35GB
Backup	x	x	x	x
Domains	1	2	5	Unlimited
Email Accounts	2	20	100	Unlimited
Servlet Contexts	1	1	10	Unlimited
Control Panel	x	x	x	x
FTP	x	x	x	x
Telnet/SSH		x	x	x
Web Mail	x	x	x	x
Web Stats		x	x	x
Perl/PHP		x	x	x
Dedicated Apache			x	x
Engine Choices	x	x	x	x
WAR/EAR	x	x	x	x
Tomcat	4.0.3	4.0.3	Latest	Latest
JBoss	2.4.1	2.4.1	Latest	Latest
Jetty			Latest	Latest

Prices, Plans, and Terms subject to change without notice. Please log on to our website for the latest Prices and Terms.

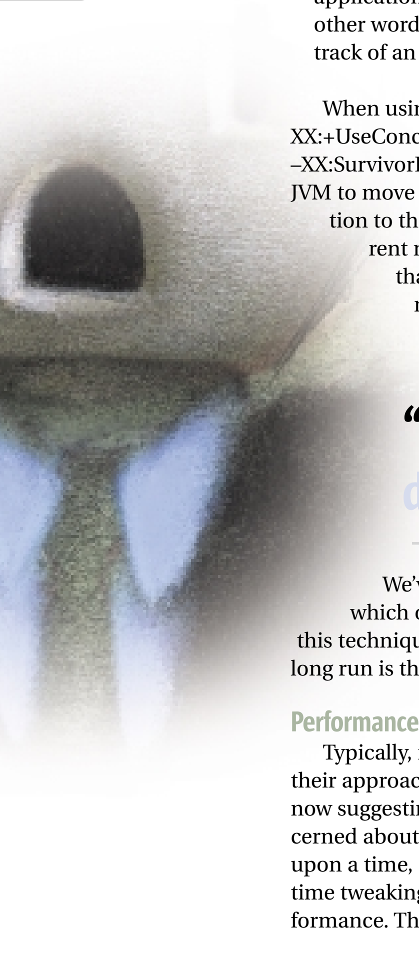
Specification for Java describes an area of memory called “Immortal Memory” where objects live for the duration of the application and garbage collection should not run.) You may try using a database; however, this may slow down your application even more. Another solution currently under the Java Community Process is JSR 107. JCache provides a standard set of APIs and semantics that allow a programmer to cache frequently used data objects for the local JVM or across JVMs. This API is still under review and may not be available yet. However, we believe it holds much promise for the Java developer community. Keep this avenue open and in mind for future architectures. What can we do now?

The pooling of objects is not new to real-time programmers. The concept is to create all your expected data objects before you begin processing, then all your data can be placed into structures without the expense of instance creation during processing time. This has the advantage of keeping your memory footprint stable. It has the disadvantage of requiring a “deep copy” method to be written to store the data into the pool. (If you simply set an object to another, you’re changing the object reference and not reusing the same space.) The nanosecond expense of the deep copy is far less than that of the object instance creation.

If the data pooling technique is combined with the proper use of the “nullify” technique, garbage collection becomes optimized. The reasons are fairly straightforward:

1. Since the object is set to null immediately after the deep copy, it lives only in the young generation portion of the memory. It does not progress into the older generations of memory and thus takes less of the garbage collector’s cycle time.
2. Since the object is nullified immediately and no other reference to it exists in some other collection object in the application, the job of the garbage collector is easier. In other words, the garbage collector does not have to keep track of an object that exists in a collection.

When using data pools, it’s wise to use the parameters “-XX:+UseConcMarkSweepGC –XX:MaxTenuringThreshold=0 –XX:SurvivorRatio=128” on the command line. These tell the JVM to move objects on the first sweep from the new generation to the old. It commands the JVM to use the concurrent mark sweep algorithm on the old generation that proves more efficient since it works “concurrently” for a multi-processor platform. For single processor machines, try the “-Xincgc” option.



“The pattern to help control garbage collection pauses most definitely improves the overall performance of the application”

We’ve seen those long garbage collector pauses, which occur after hours of execution, disappear using this technique and these parameters. Performing well in the long run is the true benefit of this last step.

Performance Results

Typically, most engineers want proof before changing their approach to designing and coding. Why not? Since we’re now suggesting that even Java programmers should be concerned about resource allocation, it better be worth it! Once upon a time, assembly language and C programmers spent time tweaking memory and register usage to improve performance. This step was necessary. Now, as higher-level object-oriented programmers we may disdain this thought. This pattern has dared to imply that such considerations, although not as low level as registers and memory addresses (instead at the object level), are still necessary for high-performance coding. Can it be true?

The underlying premise is that if you know how your engine works, you can drive it better to obtain optimal performance and endurance. This is as true for my 1985 300TD (Mercedes, five cylinder, turbo diesel station wagon) with 265,000 miles as for my Java code running on a HotSpot VM. For instance, knowing that a diesel’s optimal performance is when the engine is warm since it relies on compression for power, I let my car warm up before I “push it.” Similarly, I don’t overload the vehicle with the tons of stuff I could place in the tailgate. HotSpot fits the analogy. Performance improves after the VM “warms up” and compiles the HotSpot code into the native language. I also keep my memory footprint lean and light. The comparison breaks down after awhile, but the basic truth does not change. You can use a system the best when you understand how it works.

Our challenge to you is to take statistics before and after implementing this pattern on just a small portion of your code. Please recognize that the gain will be best exemplified when your application is scaled upward. In other words, the heavier the load on the system, the better the results.

The following statistics were taken after the pattern was applied. They are charted as:

1. Limited nullify method invocation is used where only the incoming messages are not “nullified.” (The remainder of the application from which the statistics were taken was left intact with a very lean memory usage.) There is no forced garbage collection.
2. Nullify method invocation *and* forced garbage collection is utilized.

The test environment is a Microsoft Windows 2000 X86 Family 15 Model 2 Stepping 4 Genuine Intel ~1794MHz laptop running the BEA WebLogic Server 7.0 with Service Pack 7.1 with a physical memory size of 523,704KB. The Java Message Server (JMS server), a track generator, and a tactical display are all running on the same laptop over the local developer network (MAGIC). The server makes no optimizations, even though each application resides locally. The JVMs are treated as if they were distributed across the network. They’re running on the J2SE 1.4.1 release.

The test target application is a Java Swing Tactical Display with full panning, zooming, and track-hooking capabilities. It receives bundles of tracks via the Java Message Service that are displayed at their proper location on the given image. Each track is approximately 88 bytes and the overall container size is about 70 bytes. This byte measurement does not include all the additional class information that’s also sent during serialization. The container is the message that holds an array of tracks that contains information such as time and number of tracks. For our tests, the tracks are sent at a 1Hz rate. Twenty sets of data are captured.

To illustrate the test environment, a screen capture of a 5,000 track load (4,999 tracks plus the ship) is shown in Figure

You work with objects & xml. Why doesn't your report tool?

Object Reporting

ReportMill is the world's first XML/Object Reporting System. Today's web applications are built with objects and XML - now you can organize and display the data as well formatted web pages with one line of code.

Object Reporting means:

No Redundant Database Connections
No Redundant Data Fetching
Reuse of Existing Business Logic
Reporting from Any Data Source

Dynamic PDF & Flash Output

Real web applications demand professional looking output that draws customers to your website and inspires confidence in your product. HTML screen dumps are no longer the answer.

ReportMill generates true WYSIWYG pages in several different formats - all from the same template.

PDF for paginated documents that are convenient for viewing, archiving and printing
HTML/Flash for on-screen delivery of documents that are scrollable, interactive and portable

Drag n' Drop XML sample into new document and design template

Easy Integration

Creating web pages for your application couldn't be easier. ReportMill's page layout application is powerful and easy-to-use. The developer library has basically one line of API!

+

=

Fetch Objects and generate report in PDF and Flash

View in browser, email or write to file!

Supports ColdFusion, WebLogic, WebObjects, WebSphere and any Java based application.
Visit our website to discover how easy and powerful Object Reporting can be!
Fully functional evaluation version available at **http://www.reportmill.com/webstart**
For more information, call (214) 513-1636 or send email to info@reportmill.com.

ReportMill

ReportMill Software, Inc.   Flower Mound, TX   214.513.1636   http://www.reportmill.com

1. The background shows tracks rendered with the Military Standard 2525B symbology over an image of the Middle East. The small window titled "Track Generator Desktop" is a minimized window showing the parameters of the test set through the track generator application. Notice that 45 messages had been sent at the time of the screen capture. Directly beneath this window sits the Windows Task Manager. Note that the CPU utilization is at 83%. At first this doesn't seem that bad. But at that rate, there isn't much room for the user to begin zooming, panning, hooking tracks, and so on. The final command window to the right is that of the tactical display application. The parameter "-verbose:gc" is placed on the Java command line (java –verbose:gc myMainApplication.class). The VM is performing the listed garbage collection at its own rate, not by command of the application.

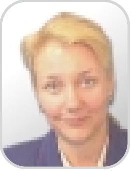
The final test of 10,000 tracks performed extremely poorly. The system does not scale; the CPU is pegged. At this point most engineers may jeer at Java again. Let's take another look after implementing the pattern.

After implementation, where the nullify methods are invoked properly and garbage collection is requested at a periodic interval (2Hz), dramatic improvements are realized. The last test of 10,000 tracks proves that the processor still has plenty of room to do more work. In other words, the pattern scales very well.

Performance Summary

The pattern to help control garbage collection pauses most definitely improves the overall performance of the application. Notice how well the pattern scales under the heavier track loads in the performance bar chart in Figure 2. The darker middle bar shows the processor utilization at each level of the message (track) load. As the message traffic increases, the processor utilization grows more slowly than without the pattern. The last light-colored bar shows the improved performance. The main strength of the pattern is how well it scales under heavy message loads.

There is another subtle strength to the pattern. This one is difficult to measure since it requires very long-lived tests. If Step 3 is faithfully followed, those horribly long garbage collection pauses that occur after hours of running disappear. This is a key benefit to the pattern since most of our applications are designed to run "forever."



Lillian Andres is a lead member of the engineering staff at Lockheed-Martin. She has 20+ years of experience and loves the challenge of not only architecting mission-critical systems but also implementing them with the latest technologies.

lillian.andres@lmco.com

Chris Cargado is a member of the infrastructure team for the U.S. Navy's "CG/DDG Open Architecture" initiative. She has spent over 12 years analyzing, architecting, evaluating, and integrating mission-critical commercial enterprise applications and military tactical weapon systems with judicious use of commercial software and hardware technologies.

chris.m.cargado@lmco.com

M. Valerie Underwood is a principal member of the engineering staff with Lockheed-Martin and is architecting enterprise computing system products while still influencing tactical system architectures. She has been working with system architectures in the challenging field of mission-critical system development for the majority of her 20-year career.

valerie.underwood@lmco.com

We're confident that many other Java applications would benefit from implementing this very simple pattern.

The steps to control garbage collection pauses are:

1. Set all objects that are no longer in use to null and make sure they're not left within some collection. "Nullify" objects.
2. Force garbage collection to occur both:
   - After some major memory-intense operation (e.g., scaling an image)
   - At a periodic rate that provides the best performance for your application
3. Save long-lived data in a persistent data area if feasible or in a pool of data and use the appropriate garbage collector algorithm.

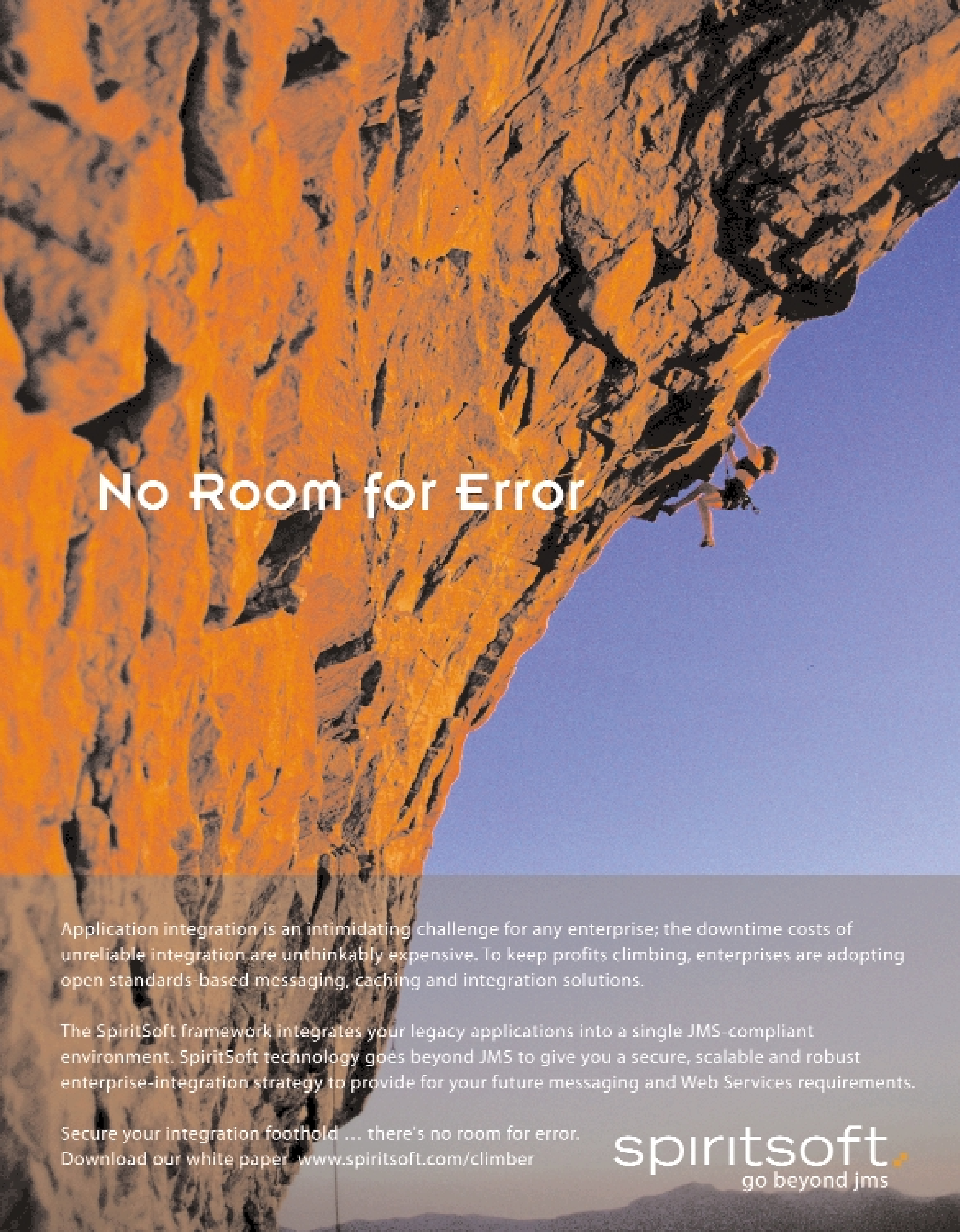
By following these three simple steps, you'll avoid those bothersome garbage collection pauses and enjoy all the benefits of the Java environment. It's time the Java environment was fully utilized in mission-critical display systems.

References

- Gupta, A., and Doyle, M. "Turbo-Charging the Java HotSpot Virtual Machine, v1.4.x to Improve the Performance and Scalability of Application Servers": http://developer.java.sun.com/developer/technicalArticles/Programming/turbo/
- *JSR 1, Real-Time Specification for Java:* http://jcp.org/en/jsr/detail?id=1
- *Java HotSpot VM options:* http://java.sun.com/docs/hotspot/VMOptions.html
- *Java Specification Request for JCache:* http://jcp.org/en/jsr/detail?id=107

Listing 1

```
public class BetterControlOfGC {
  // only schedule garbage collection for the application once
   private static boolean done = false ;

  public static void  suggestGCNow ()
  {
       System.gc () ;
  }

    private class GCStimulatorTask extends java.util.TimerTask {
      public void run () {
         suggestGCNow () ;
      }
    }

     private static GCStimulatorTask instance = null ;

     private synchronized GCStimulatorTask getInstance () {
       if (instance == null ) {
             instance = new GCStimulatorTask () ;
        }
      return instance ;
     }

    public  void scheduleRegularGC (long intervalMilliSecs ) {
     if (!done){ // only schedule 1 garbage collector per appl
        GCStimulatorTask stimulator = getInstance () ;
        java.util.Timer scheduler = new java.util.Timer () ;
        scheduler.scheduleAtFixedRate (stimulator, 10, 30

     intervalMilliSecs);
         done = true ;
      }
      else
      {
         System.err.println("GC Task already scheduled.") ;
      }
    }

     public BetterControlOfGC () {}
  }
```



Glen Cordrey
J2ME Editor

The MIDlet Marketplace

Where can you go to make your MIDlets available to the public? Let's take a brief tour of some of the Web sites that offer MIDlets to the public and explore some of the factors you should consider.

As with so much in life, one discriminator is money – are you providing your apps for free, or do you want to make money from them? Some sites, such as midlet.org, offer only free MIDlets, so if you want to make money you need to look elsewhere. Some sites provide a mix of free, shareware, and pay-per-download MIDlets, and some provide just the last.

Another factor is which devices your app targets. While "write once, run anywhere" is a nice concept, the range of physical characteristics of MIDP devices, such as screen size and black and white or color, plus whether your app uses additional APIs such as Bluetooth (JSR-82), may limit you to a subset of MIDP devices. If such limitations constrain you to a specific manufacturer or network provider, you may wish to use that vendor or provider's publication and delivery mechanism, as is available at Motorola's www.idendev.com or Cingular's http://alliance.cingularinteractive.com. However, you may not have to restrict yourself to that manufacturer's or provider's Web site(s), as some third-party sites have agreements to distribute MIDlets for manufacturers and providers. For example, Microjava (www.microjava.com) has an agreement with Motorola and Nextel that allows Microjava to certify MIDlets for Motorola/Nextel's iDen phones and distribute those MIDlets to Motorola and Nextel.

Since you probably want to expose your MIDlets to the largest possible set of potential users, place them on a site that generates a lot of traffic. Ideally you'd like to know such things as the number of visitors to the site within a specified period and the number of purchases of similar apps, but this information may not be available. So you might consider other factors instead, such as the site's target audience and how many J2ME apps it currently offers. A site that provides only J2ME apps may draw fewer visitors than one that also provides non-J2ME apps, and a large catalog of apps is likely to draw more visitors than a small catalog.

Handango (www.handango.com) has the largest number of MIDlets by far, offering over 1,700 (counting separate versions of an application for different platforms as separate applications). Because Handango also has many non-J2ME applications, your potential customer pool can include not just the J2ME cognoscenti, but also customers who are unaware of J2ME and come across your application in their quest for the latest addition to their handhelds.

Assuming you're in this for the bucks, how much can you expect to make? Although you may think your wonderful application should command premium rates, in the world of independent MIDlet marketing you need to look to volume for any real profit. In my admittedly cursory survey of pricing, most apps seem to be priced at under $7 per download, with games in the $2–$5 range and a few applications, such as browsers and e-mail viewers, in the $20–$35 range. And, of course, the distributor takes a cut – 30% at Handango and Microjava.

These are only some of the factors to consider when marketing your MIDlets, and I've only mentioned a subset of sites where MIDlets can be published. For a more extensive list visit my Web site at www.oojava.com and click J2ME->Midlet Marketplace.

• • •

Here's real news you can use: registered developers can get a free copy of Metrowerks CodeWarrior Wireless Studio from www.microjava.com or kb.motorola.metrowerks.com/motorola.

• • •

In last month's editorial I misidentified the person at NanoAmp with whom I discussed the technical aspects of their MOCA-J accelerator. Ron Stein provided me with that information, while Jason Steach arranged the discussion. My thanks to both Ron and Jason.

Glen Cordrey is a software architect working in the Washington, DC, area. He's been using Java for five years, developing both J2EE and J2ME applications for commercial customers.

glencordrey@sys-con.com

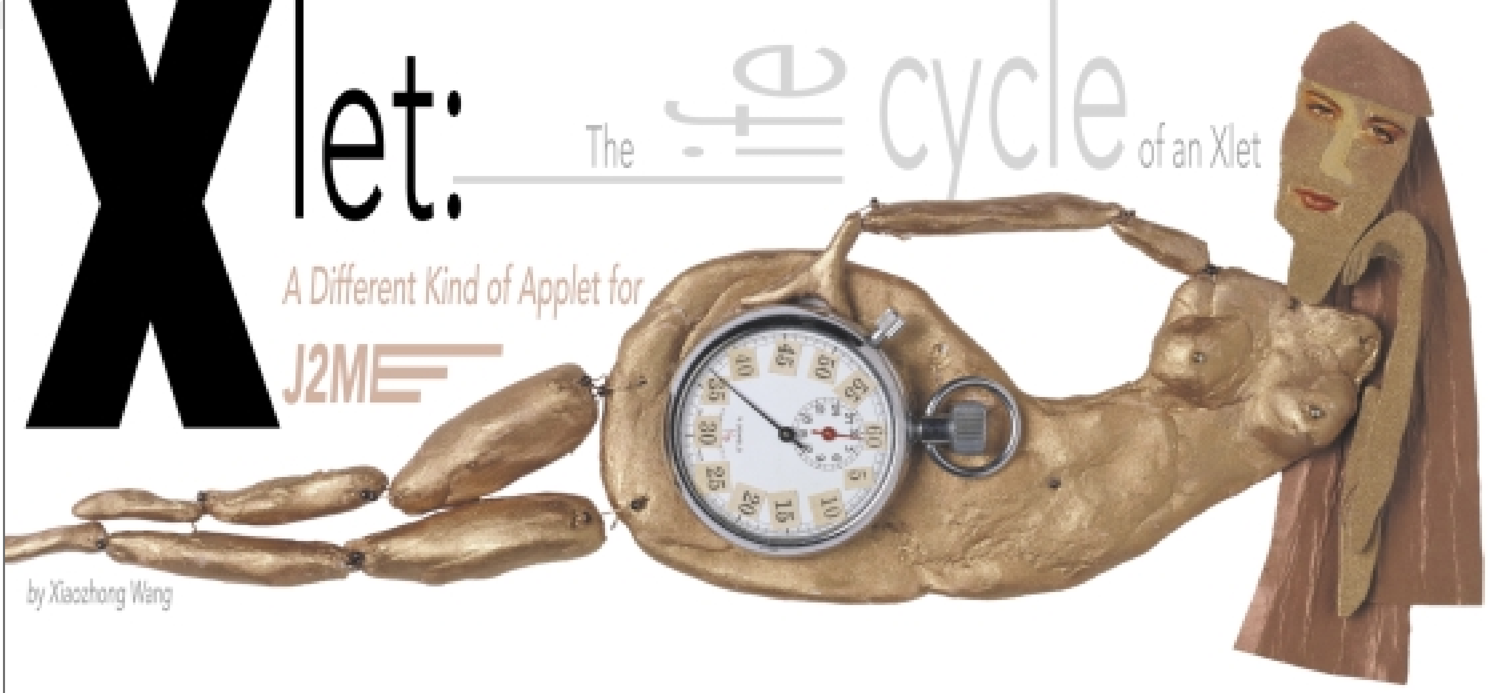
Xlet: A Different Kind of Applet for J2ME 48

The MIDlet Marketplace
Where can you go to make your MIDlets available to the public? Let's take a brief tour of some of the Web sites that offer MIDlets to the public and explore some of the factors you should consider.
46

BREW
JAVA

Xlet: The Life cycle of an Xlet

A Different Kind of Applet for J2ME



by Xiaozhong Wang

In September 2002, Sun released the J2ME Personal Profile 1.0. Unlike the MIDP, which is the core technology for Java-enabled wireless phones based on Connected Limited Device Configuration (CLDC), Personal Profile is based on the Connected Device Configuration (CDC). The CDC provides a virtual machine that includes a full Java 2 Virtual Machine feature set. Compared to CLDC, it assumes more memory and higher availability on network connections.

The Personal Profile contains a full set of AWT APIs that support a graphical user interface (GUI), including support for applets and Xlets, and provides a complete application environment for the high-end PDA market. It expands the J2ME territory to include devices that require a full GUI and a high degree of compatibility with the PersonalJava and J2SE application environment.

The Xlet application model, which is inherited from the Personal Basis Profile, is one of its most important features. What is an Xlet? Like an applet in J2SE, it's an application that must be run in an application manager. In other words, it does not have a main() method and cannot be run in standalone mode. However, it implements an interface that the application manager can use to manage its state.

Xlets potentially may play a more important role in J2ME compared to an applet in J2SE. Downloading third-party Xlets provides a way for a PDA to dynamically expand its functionality. One Xlet can even provide services to other Xlets through Inter-Xlet Communication, which makes it easy to develop client/server style applications that consist of multiple Xlets with fine modularity.

In this article, I'll talk about the Xlet life cycle and how to write an Xlet.

Xlet Life Cycle

An Xlet must implement four methods defined in the javax.microedition.xlet.Xlet interface:

```
public interface Xlet {

   public void initXlet(XletContext ctx)
      throws XletStateChangeException;

   public void startXlet()
      throws XletStateChangeException;

   public void pauseXlet();

   public void destroyXlet(boolean
          unconditional) throws
      XletStateChangeException;
}
```

Xlets, like applets, have a life cycle. The Xlet application manager uses these four methods to interact with an Xlet to manage its state. It's impractical to talk about Xlet programming without mentioning the Xlet life cycle. So before going into the programming details, let's take some time to understand an Xlet's life cycle.

An Xlet has four states:

- ***Loaded:*** The Xlet is loaded from local storage or network and its no argument constructor is called. It can enter the paused state if the Xlet's initXlet() method is called.
- ***Paused:*** The Xlet is initialized and ready to be active. It's like the ready state of a process: ready to run in the CPU at any time. It can enter the active state after the Xlet's startXlet() is called.
- ***Active:*** The Xlet is running normally. It can enter the destroyed state if its destroyXlet() method is called. It may also return to the paused state if its pauseXlet() method is called.
- ***Destroyed:*** This is the terminal state. Once it's entered, it cannot return to other states. All its resources are subject to be claimed.

In addition, an Xlet may enter the destroyed state from any other state. The possible state changes are demonstrated in Figure 1.

Implementing the Xlet Interface

Now that we have a better idea of the Xlet life cycle, let's take a closer look at the Xlet interface. The methods defined in this interface are also called life-cycle methods. Keep in

mind that user applications, including the Xlet, should not directly invoke life-cycle methods. (Even if they do, state change will not happen.) It is the Xlet application manager that invokes these methods to notify an Xlet to change its state. In a sense those methods are more like event handlers.

initXlet(XletContext ctx)

This method initializes the Xlet after it's loaded and instantiated. Note that it takes the XletContext as a parameter. This is a very important parameter as it's the only way for the Xlet to get hold of its running context. In particular, the XletContext class provides methods to retrieve the parameters that are passed to the Xlet when it's loaded and a container in which the Xlet can put AWT components. More important, through XletContext an Xlet can initiate state change itself or communicate with other Xlets.

In Listing 1, an Xlet prints all its arguments in a TextField in its initXlet() method. Note that XletContext.getContainer() may throw UnavailableContainerException. This happens if the implementation has only one displayable container and multiple Xlets are trying to access it simultaneously. In our example, upon catching such an exception the Xlet throws an XletStateChangeException to notify the Xlet application manager that it cannot initialize and needs to be destroyed.

startXlet() and pauseXlet()

The startXlet() method notifies the Xlet to start providing service and moves it to the active state. The pauseXlet() method does the opposite: it asks the Xlet to stop providing service and moves it back to the paused state.

The difference between initXlet() and startXlet() is that the former can be called only once, while the latter can be called many times in its life cycle whenever the Xlet application manager wants the Xlet to enter or resume the active state. startXlet() is a reentry method. All initialization work or things that cannot be done twice should be put in initXlet() instead of startXlet(), otherwise you risk having problems with the Xlet.

Normally, you would expect that the business logic is implemented in the startXlet() method. However, the Personal Profile specification states that all the methods defined in the Xlet interface are meant to signal state changes and the Xlet application manager expects those methods to return quickly. If a method fails to return within a certain amount of time that is implementation dependent, the Xlet manager may destroy the Xlet thinking that something is wrong with it. It's important that you don't put a lengthy implementation of business logic inside any of the life-cycle methods. If the Xlet provides some service that could go on and on, you may start a thread that provides the actual service and communicate with the thread, for example, by setting certain flags in the life-cycle methods.

"Xlets potentially may play a more important role in J2ME compared to an applet in J2SE"

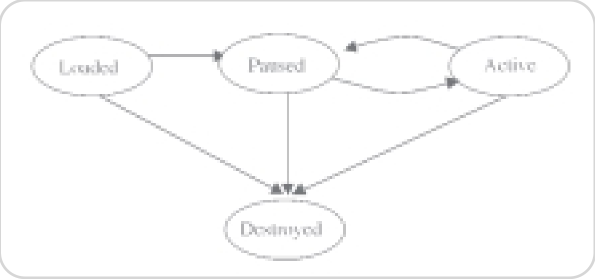


Figure 1 Possible state changes

destroyXlet(boolean unconditional)

This method signals the Xlet to terminate and enter the destroyed state. The Xlet must release all resources. The parameter "unconditional" is interesting; it's set by the Xlet application manager to signal whether it wants the Xlet to be destroyed unconditionally. If unconditional is set to false, the Xlet may throw a StateChangeException to indicate that it does not want to be destroyed. However, it's still up to the Xlet application manager to decide whether to accept or deny such a request. In other words, even though throwing such an exception is a valid response, the fate of the Xlet is still at the mercy of the Xlet application manager. If the request is accepted, the Xlet application manager will give the Xlet some time before it calls the destroy() method again, most likely with unconditional set to true. If unconditional is set to true, the Xlet application manager will ignore any XletStateChangeException and go ahead and destroy the Xlet once destroyXlet() returns.

Exceptions in Life-Cycle Methods

There are only two kinds of exceptions that can be thrown from life-cycle methods: XletStateChangeException and uncaught RuntimeException or error.

Any uncaught RuntimeException or error thrown from the life-cycle methods will immediately cause the Xlet application manager to invoke the method destroyXlet(true) of the corresponding Xlet and put it in the destroyed state. That means normally RuntimeException or error should be caught by the Xlet to prevent it from being destroyed.

On the other hand, the Xlet may intentionally throw XletStateChangeException in startXlet()to indicate that it's not ready for the state change yet.

Requesting State Change Using XletContext

As mentioned earlier, the life-cycle methods in the Xlet interface are used by the Xlet application manager to communicate to the Xlet that it wants the Xlet to change state. But what if the Xlet wants to change its own state? For example, what if an end user wants to terminate an Xlet and he doesn't want to wait for the Xlet application manager to do this?

The XletContext provides three methods that can be used by the Xlet to initiate state change: notifyDestroyed(), notifyPaused(), and resumeRequest(). notifyDestroyed() noti-

fies the Xlet application manager that the Xlet wants to terminate. Before calling notifyDestroyed(), it should release all resources, as it will in the event of a call to destroyXlet(). The Xlet unconditionally and immediately enters the destroyed state after this call.

notifyPaused() puts an Xlet in the paused state. An Xlet may call notifyPaused() to make room for other Xlets to run. Normally, a subsequent call to resumeRequest() will make the Xlet return to active state. However, as the method's name suggests, it is a request and there's no guarantee that this request will be accepted. The intended use of this method is to notify the Xlet application of its intention to return to active state. It's up to the Xlet application manager to grant and schedule such a return. It may wait for a long time before it returns, or it may not return at all; the Xlet application manager may call destroyXlet() to terminate a paused Xlet due to a lack of certain resources.

Putting It All Together: A Simple Digital Clock

Let's demonstrate the concepts and principles we've discussed so far through a sample Xlet. This Xlet displays a simple digital clock that shows the hour, minute, and second of the current time. The clock can be paused or completely terminated. I designed the Xlet in a way that the clock can be controlled not only by its own buttons, but also by the Xlet application manager via the Xlet's life-cycle methods. If you run the Xlet in Sun's XletRunner, which is the Xlet application manager for the Personal Profile reference implementation, you'll be able to control the Xlet through XletRunner's interactive menus that were originally designed to manage the Xlet's life cycle. For example, if you want to pause the clock, instead of clicking the clock's pause button, you may go to the XletRunner's "ClockXlet" menu and choose "pause". Listing 2 provides the complete code.

"There are only two kinds of exceptions that can be thrown from life-cycle methods: **XletStateChangeException** and uncaught **RuntimeException** or error"

Let's first look at the MyClock class. Its main function is to display the current time in a TextField. This class extends Thread and its main body is implemented in Thread's run() method. It also has two flags to indicate whether pause or stop is requested. If neither is requested, the clock runs normally and displays the current time once every second. If pause is requested, the clock puts itself in an indefinite wait until the pause flag is unset from outside, which awakens the thread. If stop is requested, the thread simply terminates.

MyClock is controlled by ClockXlet. The initXlet() method creates a TextField that's used by MyClock and buttons that are used to control the clock. It then starts MyClock as a thread. startXlet() and pauseXlet() methods are quite simple and return quickly. (Remember, we said that the life-cycle methods should return quickly.) They basically set or unset MyClock's pause flag, which in turn pauses or resumes the execution of the MyClock thread. The destroyXlet() method sets MyClock's stop flag, which in turn terminates the thread. This step releases resources, where the resource is the CPU.

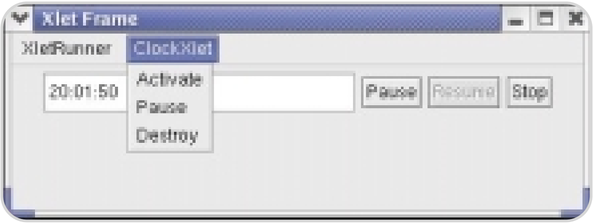


Figure 2 ClockXlet as shown in Suns's Xlet Runner

The actionPerformed() method translates the end user's actions to corresponding state change requests to the Xlet application manager. Clicking the clock's pause button will pause the clock. It will also send a request to the Xlet application manager to enter the paused state by calling XletContext.notifyPaused(). (Our ClockXlet is being nice here by not only pausing the thread, but also going to the paused state to give other Xlets a chance to run.) Clicking the clock's resume button will cause the Xlet to request a return to the active state. Once this request is accepted (hopefully!) by the Xlet application manager, it will call the Xlet's startXlet() method, where the clock will be started again. That is why we don't need to directly interact with MyClock in handling this event. Clicking the stop button will stop the clock and send a request to the Xlet application manager to terminate the Xlet.

Use the following command to run the Xlet in Sun's reference implementation (you can download Sun's Personal Profile Runtime Environment from http://java.sun.com/products/personalprofile/download.html):

```
./cvm com.sun.xlet.XletRunner \
   -name ClockXlet –path $MyClassPath
```

where $MyClassPath is the path where ClockXlet class will be loaded.

What you see on the screen will be similar to Figure 2. Use the buttons next to the TextField or "ClockXlet" menu in XletRunner to control the clock. Have fun!

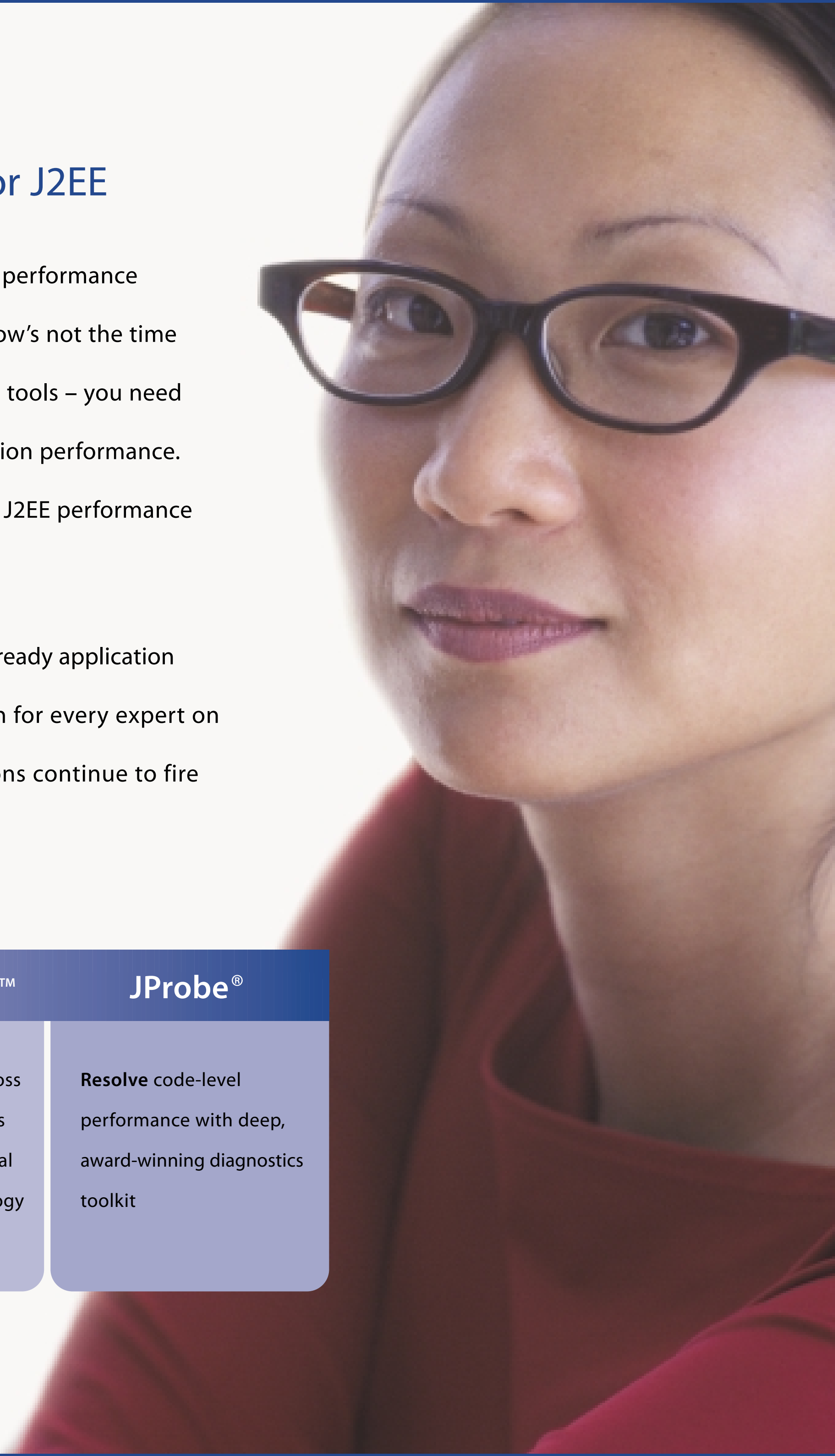
Summary

Xlets, like applets, can only run in an application manager. Xlets have a life cycle that includes four states: loaded, paused, active, and destroyed. To write an Xlet, you must implement the Xlet interface. The Xlet application manager changes the state of an Xlet through the life-cycle methods defined in the Xlet interface. Moreover, Xlets can initialize state change by using the XletContext API.

Xiaozhong Wang is a Sun Certified Java programmer and developer. Currently, he is a software engineer at Sun Microsystems. He developed the test suite for Xlet in the Personal Profile Technology Compatibility Kit 1.0, and his work has helped to refine the Xlet specification in Personal Profile 1.0.

xiaozhong.wang@sun.com

Real world performance management for live J2EE systems



Quest Central™ for J2EE

It shouldn't be happening, but it is – unexpected performance problems in your production J2EE application! Now's not the time for guesswork and random, piecemeal diagnostic tools – you need an integrated solution for managing live application performance. Accelerate detection, diagnosis and resolution of J2EE performance problems with Quest Central for J2EE.

Only Quest Central for J2EE provides production-ready application management with unparalleled diagnostic depth for every expert on the team, ensuring that your live J2EE applications continue to fire on all cylinders.

Foglight®	PerformaSure™	JProbe®
Detect critical J2EE problems 24x7 – automatically alert and trigger diagnostics	**Diagnose** problems across all tiers and components with unique transactional Tag and Follow technology	**Resolve** code-level performance with deep, award-winning diagnostics toolkit

Get the full story – attend an exclusive webinar:

http://java.quest.com/qcj/jdj

QUEST SOFTWARE®

Listing 1 Xlet argument

```
public void initXlet(XletContext ctx)
        throws XletStateChangeException {
    Container c;
    try {
        c = ctx.getContainer();
    } catch (UnavailableContainerException e) {
        throw new XletStateChangeException(e.getMessage());
    }
    TextField tx = new TextField(30);
    String[] args = (String[]) ctx.getXletProperty(XletContext.ARGS);
    String s = "";
    for (int i = 0; i < args.length; i++) {
        s = s + args[i] + " ";
    }
    tx.setText(s);
    c.setSize(200, 200);
    c.setVisible(true);
    c.add(tx);
}
```

Listing 2 A simple digital clock

```
import javax.microedition.xlet.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;

public class ClockXlet implements Xlet, ActionListener {

    TextField display;
    MyClock clock;
    Button pauseButton = new Button("Pause");
    Button stopButton = new Button("Stop");
    Button resumeButton = new Button("Resume");
    XletContext context;
    public void initXlet(XletContext ctx)
            throws XletStateChangeException {
        Container c;
        context = ctx;
        try {
            c = ctx.getContainer();
        } catch (UnavailableContainerException e) {
            throw new XletStateChangeException(e.getMessage());
        }
        display = new TextField(30);
        clock = new MyClock(display);
        pauseButton.addActionListener(this);
        resumeButton.addActionListener(this);
        resumeButton.setEnabled(false);
        stopButton.addActionListener(this);
        c.setSize(200, 200);
        c.setVisible(true);
        c.add(display);
        c.add(pauseButton);
        c.add(resumeButton);
        c.add(stopButton);
        clock.start();
    }

    public void startXlet() {
        clock.setPaused(false);
        resumeButton.setEnabled(false);
        pauseButton.setEnabled(true);
    }

    public void pauseXlet() {
        clock.setPaused(true);
        resumeButton.setEnabled(true);
        pauseButton.setEnabled(false);
   }

    public void destroyXlet(boolean unconditional) {
        clock.setStopped(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == stopButton) {
            clock.setStopped(true);
            context.notifyDestroyed();
        } else if (e.getSource() == pauseButton) {
            clock.setPaused(true);
            resumeButton.setEnabled(true);
            pauseButton.setEnabled(false);
            context.notifyPaused();
        } else if (e.getSource() == resumeButton) {
            context.resumeRequest();
        }
    }
}

class MyClock extends Thread {

    boolean paused, stopped;
    TextField display;

    public MyClock(TextField t) {
        display = t;
    }

    String getTime() {
        Calendar rightNow = Calendar.getInstance();
        String hour = String.valueOf(rightNow.get(Calendar.HOUR_OF_DAY));
        String min = String.valueOf(rightNow.get(Calendar.MINUTE));
        if (min.length() == 1) {
            min = "0" + min;
        }
        String sec = String.valueOf(rightNow.get(Calendar.SECOND));
        if (sec.length() == 1) {
            sec = "0" + sec;
        }
        return hour + ":" + min + ":" + sec;
    }

    public synchronized boolean isStopped() {
        return stopped;
    }

    public synchronized void setStopped(boolean value) {
        stopped = value;
        notifyAll();
    }

    public synchronized boolean isPaused() {
        return paused;
    }

    public synchronized void setPaused(boolean value) {
        paused = value;
        notifyAll();
    }

    public void run() {
        while (!isStopped()) {
            try {
                if (!isPaused()) {
                    Thread.sleep(1);
                    display.setText(getTime());
                } else {
                    synchronized (this) {
                        wait();
                    }
                }
            } catch (InterruptedException e) {
            }
        }
    }
}
```

ASK YOUR MANAGER

"Can we purchase the Isavix eLearning courses for our developers? They are short, high-impact, developed by industry experts, and inexpensive!"

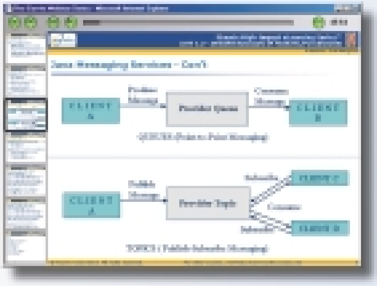
Self-service purchase, instant access ...visit http://isavix.net today!

FREE!

J2EE and .NET: A Comparison

Introduction to Web Services

Searchable Books and Certification Preparation Exams ...coming soon!


Java Messaging Services – Cont'd
CLIENT A
Provider Queue
CLIENT B
QUEUES (Point-to-Point Messaging)
CLIENT A
Provider Topic
CLIENT C
CLIENT B
TOPICS (Publish-Subscribe Messaging)

How It Works!

The Isavix High-Impact eLearning Series™ is designed to deliver online training in the most efficient manner possible. Each course is approximately one to two hours in length with no "homework" assignments. The course slides/audio can be viewed/heard online on our web site in Macromedia Flash format. In addition, the course slides can be downloaded in PDF format for printing.

Partial List of Courses:

BEA WebLogic Developer Guide!
(coming soon)

Introduction To Ant

This course will help you write your first build script, along with giving you an idea of what types of tasks Ant can do. This is a focused course, so you don't need to spend hours reading through books or online documentation to get to the heart of understanding Ant. You should go away from this course having a solid understanding of how Ant works, and how it can help you.

The Struts Framework

Struts is an open source framework useful in building web applications with Java Servlet and JavaServer Pages (JSP) technology. It you haven't used it yet, here's your chance to see why you should. This course focuses on how to use the Jakarta Struts project in your application development. Based on the Model-View-Controller pattern, the framework brings a significant advantage to your application development -- which explains the rapid growth in popularity of the Struts project.

Getting Started with XSLT

Over the last several years, XML has gained acceptance in a wide variety of business applications–but XML itself is not enough. XSLT, eXtensible Formatting Language Transformations, is one of the keys to bringing XML to life. It is a rich and versatile language that enables you to create HTML on the fly or transform business data from one XML "dialect" to another. This course will give you the essential knowledge you need to get started with XSLT.

Java Best Practices

Take your J2SE, J2EE, and J2ME skills to the next level with this informative, high-impact presentation of Java Best Practices from a noted author, mentor, and trainer.

Java has become a tremendously popular programming language in the last few years. As is the case with any new language, once the initial hype has died down and the excitement of the language's "newness" subsides, performance and efficiency become key.

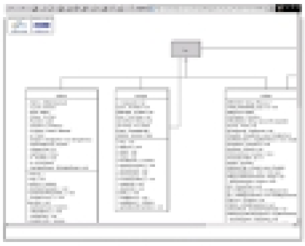
Java Performance Tuning

Developing with Java has many advantages. One of the key advantages is the capability to rapidly develop working applications. This ease of application development often comes at the expense of optimization. Developed applications are not always optimized for maximum speed and the most efficient use of memory. Over time many optimization techniques have been developed to optimize the performance of Java applications.

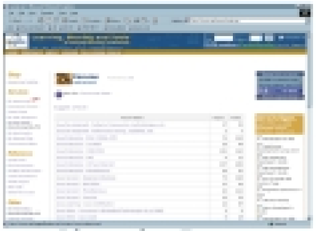
In this course you will learn many of the Java performance tuning techniques that are applied daily in the application development world. Once you become familiar with these techniques you will begin to use them instinctively as you write additional applications. Your reward will be applications that perform substantially more efficiently than un-optimized systems.

For more information, visit http://isavix.net or contact us today!

PLUS ... lots of FREE tools/resources for you!



My UML Diagrams
Class diagrams on-the-fly from you Java code!



Discussion Forums
Useful forums for J2EE and related technologies

Many more:
Code sharing, *white papers, bookmarks, secure file storage, download links, and much more...*

About Isavix

Isavix is an information-technology solutions company that provides custom services based on emerging technologies such as J2EE™, .NET™, Web Services, and many others. Since its inception in 1996, Isavix has built robust, secure, user-friendly, and cross-platform enterprise applications for many large and other leading organizations.

isavix
Delivering Emerging IT Solutions

Isavix Corp. Herndon, VA, +1 703.689.3166, info@isavix.com

Where Open Minds Meet

- **Learn** how companies have achieved higher profits and increased their productivity by utilizing Linux
- **Participate** in LinuxWorld's **world-class** education program and benefit from interactive training in the **all-new Hands-on Labs!**
- **Discover** the latest innovations and technologies from the hottest companies around
- **Hear** the latest developments and updates on the state of open source at our analyst roundtable discussion

CORNERSTONE SPONSOR

hp invent

PLATINUM SPONSORS

Computer Associates DELL IBM intel Novell ORACLE pogo [linux] Merging Business with Linux redhat Sun microsystems We make the net work. VERITAS

GOLD SPONSOR

Hummingbird

SILVER SPONSOR

APPRO HPC Cluster Solutions

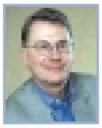
LINUXWORLD™
CONFERENCE & EXPO

Conference: August 4-7, 2003
Expo: August 5-7, 2003

The Moscone Center
San Francisco, CA

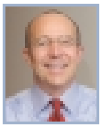
Join us this August and see why Linux is thriving! Government agencies and companies in the telecommunication, financial services, retail and manufacturing industries are turning to Linux to save money. Isn't it time you did? **LinuxWorld. Where Open Minds Meet.**

Attend Keynotes *and learn from inspiring visionaries who are devoted to Linux and open source.*



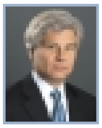
Tuesday, August 5th
10:30am-11:30am
Linux: The Next Step
Peter Blackmore
Executive Vice President
Enterprise Systems Group
Hewlett-Packard

Tuesday, August 5th
1:30pm-2:30pm
Jonathan Schwartz
Executive Vice President
Software Group
Sun Microsystems, Inc.



Tuesday, August 5th
4:30pm-5:30pm
Matthew Szulik
Chairman, Chief Executive Officer and President
Red Hat

Wednesday, August 6th
10:30am-11:30am
Linux and the Evolution of the Internet
Irving Wladawsky-Berger
General Manager
IBM Corporation



Wednesday, August 6th
3:30pm-4:30pm
Charles Rozwat
Executive Vice President
Server Technologies Division
Oracle Corporation

Special Presentation

Open to All Registered Attendees

Tuesday, August 5th
12:00pm-1:00pm

The Golden Penguin Bowl
Host: *Chris DiBona, Vice President – Marketing; Co-Founder, Damage Studios, Inc.*

Analyst Roundtable

Open to All Registered Attendees

Wednesday, August 6th
1:30pm-2:30pm

State of Open Source Roundtable
Moderator: *Larry Augustin, Partner, Azure Capital Partners*

Panelists: *Pierre Fricke, Executive Vice President of Web Application Infrastructure and Product Lifecycle Management (PLM) Infrastructure, D.H. Brown Associates, Inc.; Daniel Kusnetzky, Vice President, System Software Research, IDC; Ted Schadler, Principal Analyst in Software, Forrester; George Weiss, Vice President and Research Director, Gartner*

IDG
WORLD EXPO

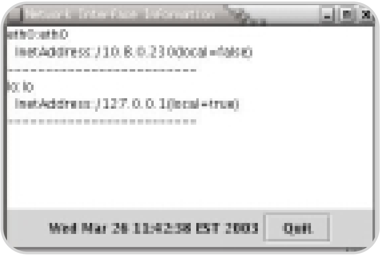
www.linuxworldexpo.com

java.net. NetworkInterface

A road warrior's friend

Duane Morin

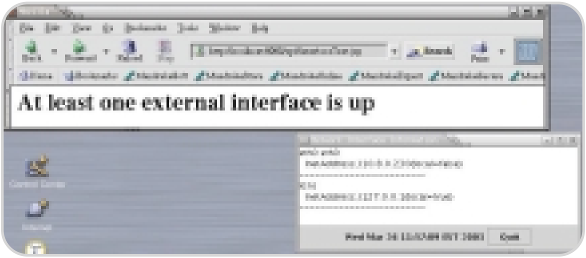
My laptop goes where I go. Some people like to read; I like to hack code. Just ask my wife – I took the laptop on vacation to the Cayman Islands. The problem I'm having is that I can never predict the state of my online connection. At home I'm wireless; at work, wired. In between, such as my 90-minute daily train commute, I'm neither. Don't get me started on the lousy connectivity from the beach. That downtime is a serious productivity killer, since all my Web-enabled code becomes unstable. There are many layers to a J2EE application these days and I don't control all of them, so I have no guarantee that my app will behave properly if a connection fails. Worst case scenarios – I might time out, hang, or crash altogether.



```
eth0:eth0
 InetAddress:/10.8.0.23(local=false)
--------------------------------
lo: lo
 InetAddress:/127.0.0.1(local=true)
--------------------------------
```

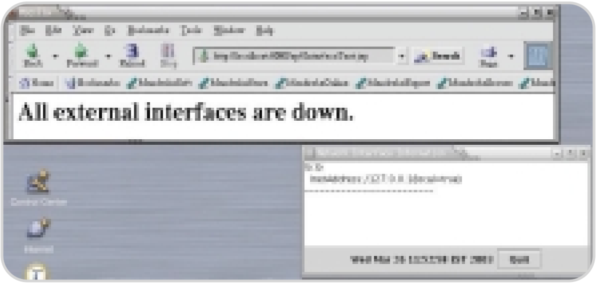
Wed Mar 26 11:42:38 EST 2003

Figure 1 Network interface information



At least one external interface is up

Figure 2 Sample tests



All external interfaces are down.

Figure 3 Sample tests

The thing is I can predict my downtimes. If I take my network card out, it's a safe bet I don't have a connection. Instead of tackling the rather large problem of detecting a good network connection, I took a different approach. I started researching whether I could tell my application that I definitely did not have a connection. If I could easily write something that says "If not connected, then don't try to perform Net-sensitive code," that would take care of the majority of my downtime. After some research I discovered that java.net.NetworkInterface, new in Java 1.4, does almost exactly what I need.

Guaranteeing that a machine is "online" is near impossible (see sidebar). Luckily I don't need to solve that problem. As I mentioned, I want to say the opposite – to tell the computer that I'm not online, rather than have it try to guess. If either of my network adapters (wireless and wired) is present and active, I can assume I have connectivity. If neither is present, or if either is present but for some reason inactive (e.g., the DHCP server hasn't given me an address or my network cable is unplugged), then I don't.

Java.net.NetworkInterface has a static method "getNetworkInterfaces" that does what I need. Listing 1 provides a test program to display a machine's interfaces. (Listings 1–4 can be downloaded from www.sys-con.com/java/sourcec.cfm.) The display refreshes every second. Figure 1 shows the program in action on my machine. I see either wlan0 (my wireless card) or eth0 (regular card). There are several static methods I could have used including getByName(String) that, if I knew what interface I was looking for, might work better for me. But why limit myself? Don't get too attached to any names you see. A Mac friend of mine reports "en0" as his network identifier. Don't limit yourself by checking for an interface because you think you know its name.

Test out the program by disabling your network adapter. On Linux, you can issue an "ifdown" command on the interface you're disabling. On Windows 2000 it's "Settings, LAN connection, Properties, Disable." The interface disappears from the screen. Pop in another card or just turn this one back on, and it should come back.

Try deactivating the connection instead. Leave the card in, but unplug the network cable. The network interface stays active, but there's no longer an InetAddress associated with it. This didn't work for me under Linux, only Win2000 (and Mac). If you are working with a DHCP server, you could also issue a manual "DHCP release" command to simulate the same effect. It isn't sufficient just to check for the existence of an interface. I must also verify that there's an associated InetAddress object.

Note: The "127.0.0.1" interface never changes. This is the local loopback device (it should say "loopback=true" on the test program), and all machines have it. It won't help you connect to the outside world. What I really want to check for is any active network interface with an associated InetAddress that is not the local loopback device. NetworkInterface doesn't have anything to help here, but InetAddress has a method isLoopbackAddress().

With algorithm firmly in hand, I wrote InterfaceCheck.java (see Listing 2). It provides a two-function API, isNetworkAvailable() and displayInterfaceInfo(). The first is where my test for good interfaces will be. The second I lifted from my earlier test program and use as a way to dump information on the available interfaces. This is just for demo purposes; it's not necessary. I deliberately kept this bean small and simple so I could use it anywhere.

The first place I applied my new test was a JSP page that reads and formats an RSS newsfeed. The taglib I'm using doesn't offer a way to fail gracefully if I ask for an input stream from an external machine and an exception is thrown. Since I don't need this particu-

Bluetooth

Enterprise Applications

802.11 3G WiFi

WLAN

Mobile Entertainment

For over seven years CTIA has fostered and helped build the wireless developer community. Today, the CTIA WIRELESS I.T. & ENTERTAINMENT 2003 show has become the one place where developers for each of the different platforms can meet to learn, exchange ideas and form crucial partnerships.

Handheld Computing

Consumer Applications

WISP

your 0000 0001 to success

Wireless Wearables

In just three days you can arm yourself with everything you need to forge ahead in the NEW wireless data industry.

Come see what CTIA WIRELESS I.T. & ENTERTAINMENT 2003 has to offer:

- Comprehensive Exhibit Floor Showcasing all of the Major Leaders in Wireless Data
- Executive Smart Pass Program on WiFi
- Quality Educational Programming and Seminars on the Latest Wireless Data Issues
- Powerful Line-up of Keynote speakers
- Key Networking Receptions

CTIA WIRELESS I.T.® & ENTERTAINMENT 2003

The NEW WIRELESS DATA INDUSTRY™

OCTOBER 21–23, 2003

Sands® EXPO and CONVENTION, Venetian Hotel, Las Vegas, NV, USA

OCTOBER 20, 2003	Pre-conference Seminars
OCTOBER 21, 2003	CTIA Educational Sessions, Special Interest Seminars & Exhibit Hall Preview Reception
OCTOBER 22-23, 2003	Exhibits, CTIA Educational Sessions [10/22 only] & Special Interest Seminars

Host

CTIA

Partners

802.11 PLANET Conference & Expo • Spring 2003

Andrew Seybold's OUTLOOK 4Mobility outlook4mobility.com

FWWF

iHollywoodForum . . . the business of interactive entertainment

MultiMeteor

Mobile Entertainment Forum

POCKET PC SUMMIT

The Wireless Authority Wireless WEEK

www.ctiashow.com

lar JSP to run properly in order to work on other aspects of the system, I wanted a way to wrap this whole section in a piece of logic that says, "If network is available, try to load this RSS feed. Else say 'Network is down' and move on."

The JSP in Listing 3 shows the result. The Net-sensitive code is surrounded by an if-then block that uses my InterfaceCheck bean to see if I'm connected. Figures 2 and 3 show some sample tests of the program in action.

Here I found a hole in my logic. My test program was working fine, and then I tried to access a URL of "http://localhost:8080/..." When my interfaces were down, it failed. But it shouldn't, since I should have access to localhost (it's my own machine after all). Listing 4 shows my final version of InterfaceCheck2.java with an extended API for this situation. Provide the address to which you want to connect; the code handles it differently if it's a local address. If an external address is provided and interfaces are down, an UnknownHostException is thrown, so return false (since the logic still holds, the URL is still not available).

I could do other things with this pro-

–continued on page 70

Duane Morin is an independent consultant with seven years of experience as a Java architect for the financial services industry. Will hack code for food, or a trip to the Cayman Islands.

dmorin@morinfamily.com

Hello, Net? Are You There?

Detecting if a machine is connected to the network is popular in any language. The problem is it makes no sense. What it often means is, can this machine use a certain port to access a certain application on a certain machine? To answer that question there's only one true test. Try it. If it works, you answered your own question.

What if it doesn't? Now you'll want to figure out where the connection failed and why. Maybe the application you wanted to connect to is not running on the destination machine. Maybe the machine is not up, period. Maybe the DNS is screwed up or a firewall is getting in the way. For all of these issues, it would still be true to say that your machine is connected to the Net. You just can't connect where you want, the way you want. When this happens, you'll want to contact the administrators of the remote machine as soon as you can, if possible. There's nothing as annoying as debugging your connection for hours before discovering that the remote machine was down for maintenance and somebody forgot to tell you.

Assuming that the remote machine is up, the safest strategy for debugging your faulty connection is to work from your own machine outward. Prove (or disprove) what you can, and move on. This article demonstrates a technique for detecting active network interfaces on a computer. No interfaces, no connectivity. If there are interfaces, move on to the next step. Usually this means "pinging" the remote machine. Unfortunately for Java programmers there's no pure solution for implementing ping, due to the low-level socket (ICMP) required. It can be done, but most Java implementations rely on JNI. If attempting to ping several remote machines returns "No route to host" (or similar), then most likely you have a gateway problem, and no traffic is getting out from your machine. Of course, if ping runs but doesn't respond, this doesn't mean anything – the remote machine can be configured to not respond to ping. Besides, since ping runs at a different level of the TCP/IP stack than most applications, you still haven't found out anything about the remote machine even if it does respond. "I can ping it but can't telnet to it" is a common bug indeed.

Everyone has different strategies for debugging a "down" machine, depending on the circumstances. The key is to work with the knowns and try to reduce the variables in the problem. Assuming you can know the status of everything between your application and the other guy's is just asking for trouble. Know what you can and can't properly test. Don't forget to call and ask if the other machine is up! Why guess when a phone call or e-mail will tell you for sure?

ExtenXLS Java/XLS Toolkit 2.1

by Extentech Inc.

Reviewed by
Peter Curran

Extentech Inc.

1032 Irving Street #910
San Francisco, CA 94122-2200
Phone: 415.759.5292
Fax: 800.787.6849
Web: www.extentech.com

Test Environment

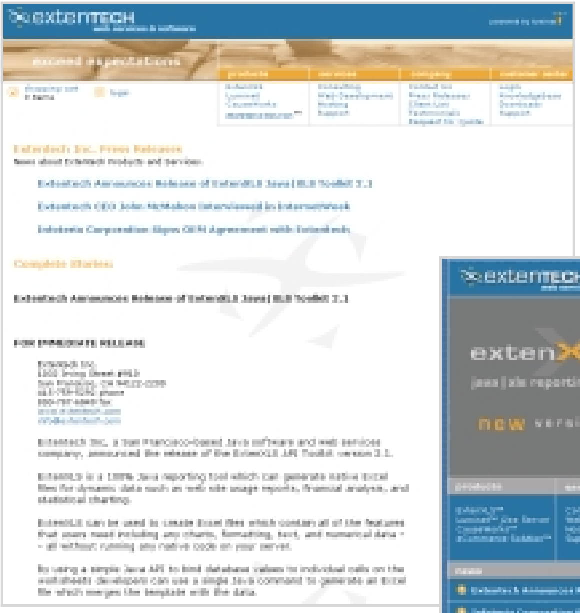
- Sun 420R, Quad 450MHz, 4GB RAM, 500GB Mounted SAN, Solaris 8
- Dell Latitude C610, Pentium 3, 833MHz, 20GB Disk, 320 MB RAM, W2K Pro SP2
- JDK 1.3.1 as well as Jython 2.1 in both cases
- Excel 2000 (9.0.4402 SR1)

For the business people of the world, Excel is like mother's milk. I'm convinced that my neighbor, a financial planner for an investment bank, does our homeowner's reconciliation for fun: a showcase for his Excel prowess. It's a sickness. Excel is powerful, simple to use, and ubiquitous in virtually every market. The problem is that those of us tasked with Excel integration know that at the binary level, Excel is a gory mess and, as a rule, does not play well with anything but COM.

Extentech offers an intuitive, pure Java API for Excel integration. Under pressure from an anxious project manager, I evaluated it side-by-side with two other Java-based Excel integration tools available on the Web: POI (Apache Software Foundation) and JExcel. The requirements were for a fast, reliable tool that could push data from a Java-based application server to heavily formatted Excel templates in either Windows or Solaris operating systems.

Extentech packages its product thoughtfully, so I was reading and writing cells within a half-hour of the download. The object model is clean, the Javadocs are fully commented, and the concise manual provides ample information about how to work through common problems. My first 30 minutes using ExtenXLS were productive and reassuring. POI, while powerful and easy on the budget, has a significantly steeper learning curve. POI's online documentation, while amusing and voluminous, is comparatively arcane. Extentech got me started much faster – a huge plus when you're strapped for time.

ExtenXLS works by first ingesting the Excel spreadsheet from either a byte array, file path, or InputStream, then parsing the binary spreadsheet and providing an API for accessing Workbook, Spreadsheet, Row, Cell, Formula, and other normal Excel objects. Once changes are written in memory through the API, the spreadsheet can then be stored back in its original form.



Extentech Inc. Press Releases
News about Extentech Products and Services

Extentech Announces Release of ExtenXLS Java | XLS Toolkit 2.1
Extentech CEO John McMahon Interviewed in InternetWeek
Infoteria Corporation Signs OEM Agreement with Extentech

Complete Stories

Extentech Announces Release of ExtenXLS Java | XLS Toolkit 2.1

FOR IMMEDIATE RELEASE

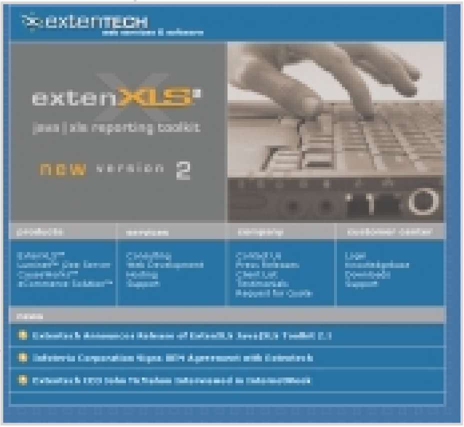
Extentech Inc.
1032 Irving Street #910
San Francisco, CA 94122-2200
415-759-5292 phone
800-787-6849 fax
www.extentech.com
info@extentech.com

Extentech Inc., a San Francisco-based Java software and web services company, announced the release of the ExtenXLS API Toolkit version 2.1.

ExtenXLS is a 100% Java reporting tool which can generate native Excel files for dynamic data such as web site usage reports, financial analysis, and statistical charting.

ExtenXLS can be used to create Excel files which contain all of the features that users need including any charts, formatting, text, and numerical data -- all without running any native code on your server.

By using a simple Java API to bind database values to individual cells on the worksheets developers can use a single Java command to generate an Excel file which merges the template with the data.



Extentech Announces Release of ExtenXLS Java | XLS Toolkit 2.1
Infoteria Corporation Signs OEM Agreement with Extentech
Extentech CEO John McMahon Interviewed in InternetWeek

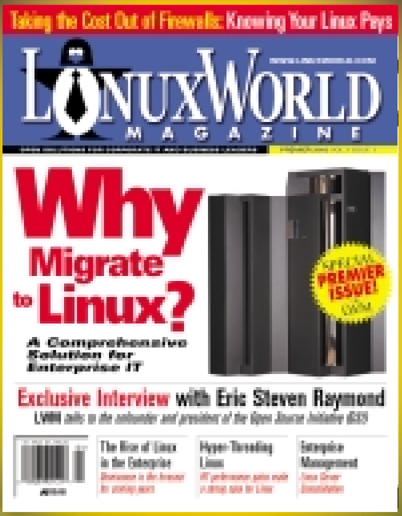
```
//Construct a workbook from a path string
String str_fileNameIn = "simple.xls";
WorkBookHandle book = new
WorkBookHandle(str_fileNameIn);
WorkSheetHandle sheet =
wbh_bookIn.getWorkSheet("Sheet1");
CellHandle cell = sheet.getCell("A1");

//Reading the value of an existing cell by ID
String s = (String) cell.getStringVal();
System.out.println("Cell G8: " + s);

//Writing the value of a cell
cell.setVal("Hello Darlin' …");

//writing back to file
byte foo[] = book.getBytes();
File file_Out = new File(str_fileNameIn);
FileOutputStream fileOS_fileoutputstream = new
FileOutputStream(file_Out);
fileOS_fileoutputstream.write(foo);
fileOS_fileoutputstream.close();
```

Code Sample 1: A very simple example of opening, reading, and writing to/from a file on disk

The key differentiator that sold us on ExtenXLS was its ability to write to spreadsheets that contained macros. All other Excel integration products that I've seen truncate macros and VBA code, no matter how simple, and write only data back to the spreadsheet, rendering it useless and/or corrupt! With POI, I found that files with macros would decrease in size after write operations by about the same number of bytes as I had macro code. Subsequent attempts to open the file would generally fail. ExtenXLS hiccupped on only

Premiering
August
2003

www.LinuxWorld.com


Taking the Cost Out of Firewalls: Knowing Your Linux Pays

LinuxWorld Magazine

Why Migrate to Linux?

A Comprehensive Solution for Enterprise IT

Special Premier Issue!

Exclusive Interview with Eric Steven Raymond

The Rise of Linux in the Enterprise

Hyper-Threading Linux

Enterprise Management

The Leading Magazine for Corporate and IT Managers

LinuxWorld Magazine

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *LinuxWorld Magazine* is aimed squarely at providing this group with the knowledge and background that will allow them to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *LinuxWorld Magazine* will not feature low-level code snippets but will focus instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month will see a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features will include:

Advice on Linux Infrastructure

Detailed Software Reviews

Migration Advice

Hardware Advice

CEO Guest Editorials

Recruiting/Certification Advice

Latest News That Matters

Case Studies

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY

$49^{99}

12 ISSUES/YR

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE*

SUBSCRIBE TODAY!

WWW.SYS-CON.COM
OR CALL
1-888-303-5282

FOR ADVERTISING INFORMATION:
CALL 201 802.3020 OR
VISIT WWW.SYS-CON.COM

SYS-CON MEDIA
The World's Leading i-Technology Publisher

the most Byzantine spreadsheets I tried, and was polite enough to throw a comprehensible exception.

When I first evaluated ExtenXLS in Q4 2002, I had two complaints: no InputStream constructor (only files and byte arrays) and no support for named ranges. The InputStream constructor was provided as a patch release within days of our enhancement request, and named range support was recently announced as a new feature.

For our purposes, these two improvements have been huge. The InputStream allows us to take spreadsheets directly from the application server document store, manipulate them without any disk I/O, and stream them back to the document store. Named range support abstracts spreadsheet data from its location within the spreadsheet – our customers are free to change their spreadsheet layout without impacting the application server integration. If the customer wants to put the task percentage complete field in D8 rather than D9, the application integration is not impacted.

Performance improvements have been noticeable as well. ExtenXLS version 1.4 took up to 30 seconds to ingest our larger spreadsheets, whereas version 2.0 does the same job in under three. Virtually all of the overhead now comes from our own business logic.

The chief criticisms I have now are bugs, not feature deficiencies. Occasionally I find that template formatting, such as boxes around certain regions, colored regions, etc., is destroyed by writes to adjacent cells. We surmounted these problems by laying out the templates more strategically, and by educating our users on some of the fussy details.

Customer licensing is simple to understand – being based on the number of CPUs in the deployment at $1,145 per CPU. Deployment licenses come with installation support (not that you would need it), and one developer seat per CPU. Developer licenses can be purchased independently, and are also reasonably priced at $150.

In my view, ExtenXLS faces two challenges going forward. First, the Apache Software Foundation produces excellent products that are widely adopted in the Java community. Luckily for Extentech, customers are still willing to pay a premium for dedicated support, and the ExtenXLS product is easily as good as POI, and in my view, even better.

More important, however, Extentech, like any software vendor, needs to look carefully at its Microsoft strategy. Following Sun's lead with an all XML-based office suite in StarOffice 6, Microsoft has used XML under the covers in Office 2003, making the novelty of a Java Excel parser much less novel. Nevertheless, the release of Office 2003 and the adoption of it in the enterprise are two very different things. Extentech has the interim to formulate new, fast, reliable, feature-rich, and well-packaged ways of bridging the .NET and Java worlds.

Peter Curran, a software architect for Intraspect Software of Brisbane, California, builds collaborative applications for high-tech vendors, investment banks, and systems integrators. The views expressed herein are those of the author and not necessarily endorsed by his employer.

pcurran@intraspect.com

Snapshot

Target Audience: Developers, architects, and analysts

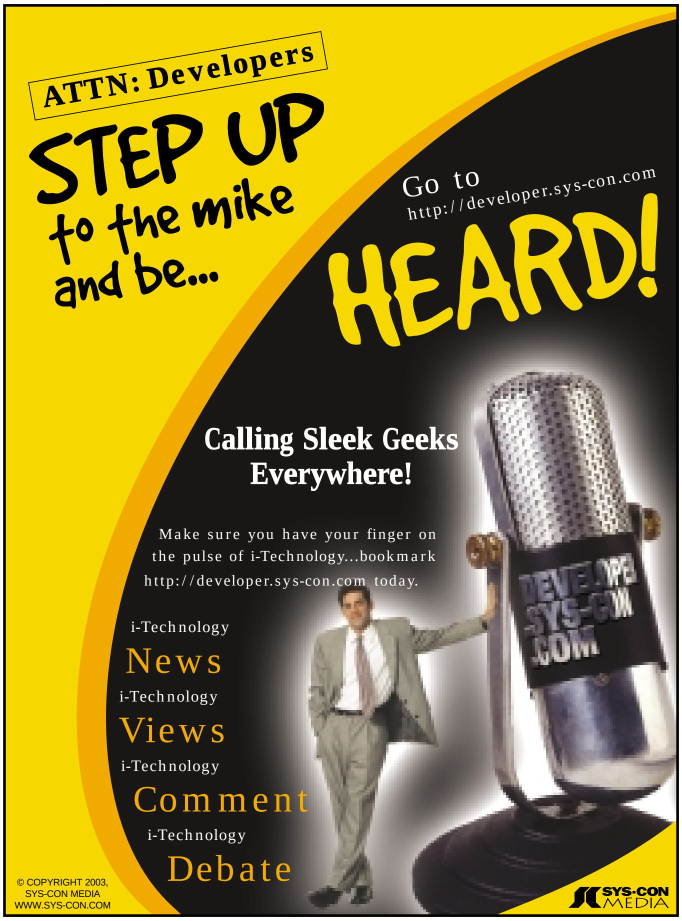
Level: Beginner to intermediate

Pros:

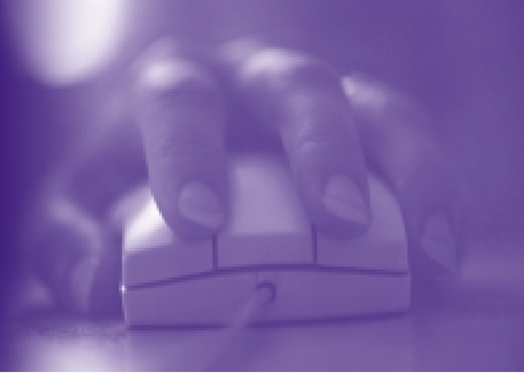
- Intuitive, flexible, well-documented API
- Can read/write spreadsheets that contain macros and VBA
- Timely, thorough support
- Fast, reliable

Cons:

- Some difficulty with extremely complicated spreadsheets
- Occasional formatting problems



SaturdaySessions



We understand the pressures of work and how difficult it can be to get time off. That is why we have designed this workshop to be held in one day and, as a special bonus, on the weekend, so no days off from work. **Your boss will be happy!**

Alan Williamson
JDJ Editor-in-Chief

M	T	W	T	F	S	S

Coming to you...

May:	June:
NEW YORK	ATLANTA
DALLAS	BOSTON

JDJ Workshop
with Alan Williamson

Performance > Efficiency > Reliability

This one-day intensive workshop is designed for developers who wish to increase the efficiency and reliability of their code development.

1) The day will begin by looking at the various hints and tips you can utilize at the code level to improve the quality and reduce the number of bugs you have to contend with.

2) The next part will look at Apache's Ant and how you can use this freely available tool for your own development, irrespective of your IDE.

3) Last, and most important, as the old saying goes: "You can never do enough testing." This session will look at JUnit and show you how to start building test harnesses for your code so you can begin your testing strategy.

>Performance

Java is a powerful language. While it offers a rich array of tools, the fundamentals mustn't be overlooked. Improving your code at the core layer will result in great improvements in efficiency and produce (hopefully) less bugs. We'll look at the do's and don'ts of programming and learn many hints and tips that will accelerate your Java coding.

>Efficiency with Ant

Apache's Ant is a powerful scripting tool that enables developers to define and execute routine software development tasks using the simplicity and extensibility of XML. Ant provides a comprehensive mechanism for managing software development projects, including compilation, deployment, testing, and execution. In addition, it is compatible with any IDE or operating system.

> Reliability with JUnit

A critical measure of the success of software is whether or not it executes properly. Equally important, however, is whether that software does what it was intended to do. JUnit is an open-source testing framework that provides a simple way for developers to define how their software should work. JUnit then provides test runners that process your intentions and verify that your code performs as intended. The result is software that not only works, but works in the correct way.

What you will receive...

- ✓ INTENSIVE ONE-DAY SESSION
- ✓ DETAILED COURSE NOTES AND EXCLUSIVE ONLINE RESOURCES
- ✓ JDJ CD ARCHIVE

ONLY $295*

*JDJ subscribers
($395 non-subscribers)

*GROUP DISCOUNTS AVAILABLE

To Register

www.sys-con.com/education

Call 201 802-3058

SPONSORED BY
JAVA DEVELOPER'S JOURNAL

PRODUCED BY
SYS-CON EVENTS

PRESENTED BY
SYS-CON i-TECHNOLOGY EDUCATION

ALL BRAND AND PRODUCT NAMES USED ON THIS PAGE ARE TRADE NAMES, SERVICE MARKS, OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

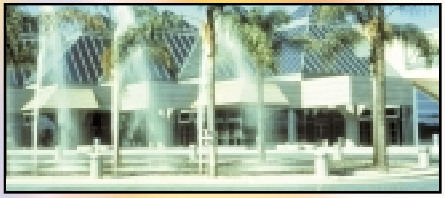
International Web Services Conference & Expo

Web Services Edge WEST 2003

web services EDGE conference & expo

SEPT. 30 - OCT. 2, 2003

Santa Clara, CA



EXTENDING THE ENTERPRISE WITH WEB SERVICES THROUGH JAVA, .NET, WEBSPHERE, MAC OS X AND XML TECHNOLOGIES

Featured technologies and topics will include:

- Focus on Java
- Focus on Web Services
- Focus on Mac OS X
- Focus on XML
- Focus on .NET

XML

WebSphere

Microsoft .net

Mac OS X

COMING IN 2004

web services EDGE conference & expo

BOSTON

FEBRUARY 24-26, 2004

Over 100 participating companies will display and demonstrate over 300 developer products and solutions.

Over 2,000 Systems Integrators, System Architects, Developers, and Project Managers will attend the conference expo.

Over 60 of the latest sessions on training, certifications, seminars, case studies, and panel discussions will deliver REAL World benefits, the industry pulse and proven strategies.

CONTACTS:

Exhibit & Sponsorship	Conference & Education
GRISHA DAVIDA	MICHAEL LYNCH
201 802-3004	201 802-3055
grisha@sys-con.com	mike@sys-con.com

www.sys-con.com

Java is a registered trademark of Sun Microsystems, .NET is a registered trademark of Microsoft, Mac OS X is a registered trademark of Apple Computer, Inc., WebSphere is a registered trademark of IBM. All other product names herein are the properties of their respective companies.

Web Services Edge WEST 2003

WEB SERVICES EDGE CONFERENCE & EDUCATION

web services EDGE conference & expo

SEPT. 30 - OCT. 2, 2003

Santa Clara, CA

JAVA TECHNOLOGY

The Java Track features presentations aimed at the beginner, as well as the seasoned Java developer. Sessions will explore the whole spectrum of Java, focusing on J2EE, application architecture, EJB & J2ME. In addition the Track will cover the latest in SWT, Ant, JUnit, open source frameworks, as well as an in-depth look into the vital role that Java is playing in building and deploying Web services.

Sessions will focus on:

- Enterprise Java 1.4
- Ant Applied in "Real World" Web Services
- Developing Application Frameworks w/SWT
- Empowering Java and RSS for Blogging
- JUnit: Testing your Java w/JUnit
- JDK1.5: The Tiger
- Simplifying J2EE Applications
- Using IBM's Emerging Technologies Toolkit (ETTK)
- Apache Axis
- Meeting the Challenges of J2ME Development
- Integrating Java + .NET
- Squeezing Java

.NET TECHNOLOGY

Presentations will explore the Microsoft .NET platform for Web services. To the average developer, it represents an entirely new approach to creating software for the Microsoft platform. What's more, .NET development products - such as Visual Studio .NET - now bring the power of drag-and-drop, GUI-based programming to such diverse platforms as the Web and mobile devices.

Sessions will focus on:

- ASP.NET
- Security
- VB.NET
- .NET and XML
- Smart Device Extensions for VS.NET
- Best Practices
- Shared Source CLI
- .NET Remoting
- Smart Devices in Health Care Settings
- Mobile Internet Toolkit
- ROTOR
- Portable .NET
- ASP.NET Using Mono
- Using WSE with IBM's WSTK
- GUI applications Using Mono
- Portals – Windows Sharepoint Services/Sharepoint Portal Server
- Windows Server 2003 and IIS 6
- .NET and Java Interoperability
- Distributed .NET for Financial Applications
- Developing C# with Eclipse

Microsoft .net

WEB SERVICES TECHNOLOGY

Presentations will include discussions of security, interoperability, the role of UDDI, progress of the standards-making bodies, SOAP, and BPM. Case studies cover the design and deployment of Web services in the marketplace.

Sessions will focus on:

- Interoperability
- Enterprise Networks
- Web Services Management
- Web Services Standards
- Web Services Orchestration
- Security (WS-Security, SAML)
- BPEL4WS
- UDDI: Dead or Alive?
- ebXML & Web Services
- EAI & Web Services
- RPC vs. Messaging: Uses and Differences
- User Interfaces for Web Services
- Web Services Best Practices
- Service Oriented Architecture

MAC OS X

OS X represents a new wave of operating systems. It combines the ease of use of a Mac with the power of Unix. Sessions in this track will highlight the use of the Mac OS X platform in applications and Web services development, deployment and management.

Sessions will focus on:

- Introducing OS X (Panther): What's New?
- Quick Applications using AppleScript
- Enterprise Java and OS X
- Developing Web Services Using WebObjects
- Xserve: Ease of OS X and Power of Unix
- Introducing Quartz: 2D Graphics for Apple
- OS X for the Unix Developer
- Securing OS X Applications
- Java and OS X: A Perfect Marriage
- Programming Rich User Interfaces Using Cocoa

X Mac OS X

XML TECHNOLOGY

Presentations will focus on the various facets of XML technologies as they are applied to solving business computing problems. Sessions will include emerging standards in XML Schemas, XML repositories, industry applications of XML, applying XML for building Web services applications, XML/XSLT/XQuery-based programming using Java/.NET, XML databases, XML tools and servers, XML-based messaging, and the issues related to applying XML in B2B/EAI applications. The XML Track is geared for audiences ranging from beginners to system architects and advanced developers.

XML

Sessions will focus on:

- XML Standards & Vocabularies
- Introduction to XForms
- Securing Your XML and Web Services Infrastructure
- XQuery Fundamentals: Key Ingredient to Enterprise Information Integration
- XML and Enterprise Architecture: Technology Trends
- Standards-Based Enterprise Middleware Using XML/Web Services
- XML and Financial Services
- Canonical Documents for Your Business: Design Strategies
- XPath/XSLT 2.0: What's New?
- XML Schema Best Practices
- XML in EAI, Enterprise Portals, Content Management

For more information visit **www.sys-con.com** or call **201 802-3069**

PRODUCED BY SYS-CON EVENTS

MEDIA SPONSORS:

WebServices Journal · JAVA Developer's Journal · WebSphere Developer's Journal · XML Journal · .NET Developer's Journal · BEA WebLogic Developer's Journal · LinuxWorld Magazine · wireless Business & Technology · Linux Business Week · CF Advisor · ColdFusion Developer's Journal · PowerBuilder Developer's Journal · market wire · HSP Street.com · OASIS · SD Times · LinuxWorld.com · SAMS · Real Industry

Industry News

Fiorano Releases FioranoMQ 7.0

(*Los Gatos*) – Fiorano Software, Inc., a provider of enterprise integration middleware has announced the release of FioranoMQ 7.0, the next generation of the industry's Java messaging server. New features include support for high availability (HA), distributed transactions (XA), and enhancements to the administration and management tools.
www.fiorano.com

Wily Technology Integrates Introscope 4.1 with Oracle9*i*AS

(*San Francisco*) – Wily Technology, a provider of Enterprise Java Application Management and a member of the Oracle PartnerNetwork, has announced that Introscope 4.1, the newest version of Wily's Enterprise Java Application Management solution, is now integrated with and supports Oracle9*i* Application Server. Together, these solutions help businesses ensure that their Java applications on Oracle9*i* Application Server can meet the high demand for performance and availability.
www.wilytech.com

Oak Grove Systems Introduces New Managed Source Licensing Model

(*Calabasas, CA*) – Oak Grove Systems, the business process execution company, has announced the release of Reactor 5 Source Code PLUS, a Java-based product that will provide developers with a low-risk way to process-enable applications. Reactor 5 Source Code PLUS Managed Source licensing model provides all the benefits of licensed software: embedding services, full support, and ongoing development, along with the freedom of open source; and royalty free distribution and ownership of the completed software product code "to the core."
www.oakgrovesystems.com

Sun Launches Products and Programs to Unite Wireless Java Community

(*San Francisco*) – Sun Microsystems, Inc., has launched a comprehensive mobile developer program and joined forces with wireless industry leaders to announce an industry initiative for application testing to help the wireless industry bring Java technology-based applications and services to market quickly and cost effectively. Sun also announced plans to launch an end-to-end mobile enterprise development platform to help service providers and enterprises extend their current IT investments into 2.5 and 3G wireless networks.
www.sun.com

Oracle Previews New Approach to Application Development

(*San Francisco*) – Oracle has introduced Oracle9*i* JDeveloper version 9.0.5. The new release simplifies application development and improves developer productivity through its new Oracle Application Development Framework (Oracle ADF).

Oracle ADF is a standards-based J2EE framework that provides a foundation for designing and creating J2EE applications and Web services for developers of all skill levels, allowing them to choose the technologies and development style that best match their skill sets and the requirements of their specific projects.
www.oracle.com

New Parasoft Java Solution Debuts at JavaOne

(*Monrovia, CA*) – Parasoft, a provider of Automated Error Prevention software and solutions, has announced the general availability of Parasoft Java Solution, a complete package of automated tools, services, and the best practices needed to prevent errors in Java applications. The Parasoft Java Solution integrates error prevention and monitoring techniques into the full life cycle of any Java development project.
www.parasoft.com

ObjectFX Delivers SpatialFX 3.3

(*Minneapolis*) – ObjectFX is now delivering version 3.3 of its flagship software platform SpatialFX. The updated product adds new features targeting government customers along with advanced Web capabilities and advanced J2EE capabilities and conformance. ObjectFX is a provider of location-based services (LBS) software based on J2EE standards that supports asset management capabilities through the integration of business information with Web-enabled spatial interfaces such as mapping, vehicle routing, and other spatial operations.
www.objectfx.com

Evant Announces Support for IBM WebSphere Application Server

(*San Francisco*) – Evant, a provider of retail management software and services, has announced support for IBM WebSphere Application Server, a high-performance and scalable transaction engine for e-business applications based on the J2EE standard. Evant's support for WebSphere signals an increase in its activities with IBM. Evant is already a participant in IBM's Initiative for Emerging Technology Developers, a program that nurtures innovative ISVs (Independent Solution Vendors) in hot technology growth areas by providing full technical and marketing support to help them succeed.
www.evant.com

Borland/Sony Ericsson Team to Accelerate Java Development for Wireless Devices

(*San Francisco*) – Borland Software Corporation and Sony Ericsson Mobile Communications AB have announced a strategic relationship to extend the support of enhanced mobile application development. Under the terms of the agreement, Borland plans to distribute and support the Sony Ericsson Java Software Development Kit (SDK) through the Borland JBuilder 9 Mobile Edition, accelerating the application development life cycle for wireless Java-based devices.
www.borland.com
www.ericsson.com

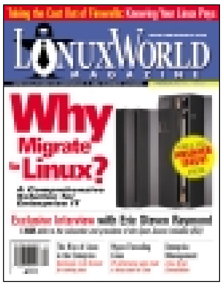
Macromedia ColdFusion MX 'Java Verified' for Portability Across J2EE Application Servers

(*San Francisco*) – Macromedia has announced that ColdFusion MX has achieved "Java Verified" status under the Sun Microsystems Java Verification Program. The Java Verification Program is designed to identify enterprise applications developed with J2EE technology that are intended to be portable across different implementations of J2EE. Macromedia ColdFusion MX, the server scripting environment for creating Internet applications, brings the ease of use and productivity of ColdFusion to the J2EE platform.
www.macromedia.com

A LIMITED TIME SAVINGS OFFER FROM SYS-CON Media

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!



TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

Pick a 3-Pack, a 6-Pack or a 9-Pack

□ 3-Pack	□1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 6-Pack	□1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 9-Pack	□1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.

□ **Linux World Magazine**

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Java Developer's Journal**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **XML-Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **WebLogic Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ **Wireless Business & Technology**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **WebSphere Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY
WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $275⁰⁰ Pay only $175 for a 1 year subscription plus a FREE CD

- 2 Year – $299.00
- Canada/Mexico – $245.00
- International – $315.00

6-Pack

Pick any 6 of our magazines and save up to $350⁰⁰ Pay only $395 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $669.00
- Canada/Mexico – $555.00
- International – $710.00

9-Pack

Pick 9 of our magazines and save up to $400⁰⁰ Pay only $495 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $839.00
- Canada/Mexico – $695.00
- International – $890.00

Subscribe Online Today www.sys-con.com/2001/sub.cfm

SYS-CON MEDIA

Letters to the Editor

Test Infected Code

I think Jason Bell is right in his editorial "Testing, Testing…" (Vol. 8, issue 6) that the secret to testing is adopting the right mind set. I agree with Kent Beck and others that the best way to write test code is to write the test before the code you're going to test. This might sound crazy at first but it makes a lot of sense. First, it makes you more honest. If you've already written and compiled code you tend to lose interest. The fun bit of working out how to do it and making it compile and run is over. Writing a test is boring after that. Writing the test first is just as challenging as writing the code though, and you can be honest and imaginative thinking up test cases because you haven't invested any time or pride in what you are testing yet.

Ben Butchart
b.butchart@cs.ucl.ac.uk

Tester = Lowlife? Are You Working for MS?

I cannot argue with the politics of the subject, which sounds unfortunately true ("And the Artificial Stupidity Award Goes to… [Vol. 8, issue 6]). However, I object to the notion that art school rejects are "forced to take up jobs as software testers, technical writers, or quality auditors."

I'm a developer at heart. After several years and several successful projects, I did a few stints as a technical writer and spent two years as a software tester. In both cases, the new positions involved a significant financial bonus. These jobs were by no means demotions.

I know that in a lot of places, the tech writers and the software testers couldn't code their way out of a wet paper bag, and are considered oxygen wasters by the coders. The attitude of coders toward doc and test *has* to change. Otherwise, IT as a whole will go into a deep crisis.

Fred Mora
via e-mail

Java an Open Standard?

Sun needs to decide if they want J2EE (and Java for that matter) to be an open standard or a product ("Pulling at a Thread" by Alan Williamson [Vol. 8, issue 5]). This continuing mix-up of commercial interests with compliance questions is bad for everyone including Sun.

Alexander Jerusalem
ajeru@vknn.org

Testing Is Vital

I am a bit surprised that the word *test* was not mentioned in Ajit Sagar's editorial about performance "The Proof Is in the Concept" (Vol. 8, issue 5). Perhaps test is included in the POC resources that it mentions the client is making available. I don't think you need to understand the particular definitions or distinctions between the proof-of-concept and a prototype that were given, but at whatever stage of the project you should have some customer performance requirements that have been expressed somehow (otherwise why bother considering performance) for which you can define and execute a test. My suggestion is to consider testing for the vital few performance requirements from the beginning of the design as that will help you produce a better system sooner. Thanks for the article.

Chris Thompson
cthompson@empirix.com

Increase Agility

Sun in this scenario should stick to its current approach ("Is J2EE Too Big for Its Own Good?" by Nigel Thomas [Vol. 8, issue 4]). Remember that application servers are not bought by developers who understand the various specifications. Instead they're usually purchased by a corporate procurement person or nontechnical manager. Giving them the opportunity to search for a single identifier increases agility.

James McGovern
james@webservicesarchitecture.com

Swing's Time Is Over

SWT is really beneficial ("SWT: A Native Widget Toolkit for Java" by Joe Winchester and Steve Northover [Vol. 8, issue 5]). Since the Eclipse team ports the native implementation to multiple platforms (Linux, AIX, Windows, to name a few) the toolkit is just about as portable as Java AWT and Swing, with much better performance. The performance is the key, especially with large applications. I have run Eclipse on AIX (a horrible OS) and it runs faster than smaller Java apps, and like the last guy said: if the user can't tell it's a Java app, all the better. Stability and performance, that's what we need from a widget toolkit. Though I like the architecture of Swing, I think that its time is over.

Tim Osten
tosten@nims.net

java.net. NetworkInterface

–continued from page 60

gram. You probably noticed that the logic is called every time. The most obvious thing would be to cache the results and update periodically. There's something elegant about leaving it small and simple, however. You can drop it in anywhere you like (JSP, custom tag, Swing application, etc.) and have it work as is. Why keep messing with it?

Conclusion

This technique doesn't detect good connections; it detects their absence. It's easier. Your connection could fail for any number of unpredictable reasons on either end or somewhere in the middle. But in situations where you know you don't have any connectivity and temporarily need to tell your code not to even try it, NetworkInterface handles the job nicely.

From Within the Java Community Process Program

More mobility, less complexity

Onno Kluyt

Welcome to the July edition of **JSR Watch**! Each month this column provides information about the JCP program: newly submitted JSRs, new draft specs, Java APIs that were finalized, and other news from the JCP program. This month's column discusses a set of new J2EE technology JSRs, and a scripting JSR, but I'll start off with the J2ME environment.

Two J2ME JSRs Are Nearly Final

JSR 179, Location API for J2ME technology, specifies how to write mobile location-based applications for devices with limited resources, producing information to an application about the device's physical location. This effort, led by Nokia, successfully navigated the final approval ballot on June 2, collecting 14 yes votes from the ME EC members. The second one is JSR 195, Information Module Profile. This JSR just posted its proposed final draft and is expected to enter the final approval ballot very shortly. Originally submitted in October of 2002, it has been moving through the community process at a very decent clip. The effort is closely related to MID-P in that it provides a Java runtime environment similar to MID-P but for devices that don't have graphical display capabilities. The JSR is co-led by Siemens and Nokia.

Ease of Development

If you attended this year's JavaOne conference, you'll have seen the above phrase featured in various technical sessions. It is also a topic that the Executive Committees are discussing at our monthly meetings, where it has the ominous title "Making Java more attractive." These meetings contemplate whether the community is developing the right technologies and the right tools, and has the needed developer programs to make Java developers more productive and significantly increase their number beyond the current three million. The JavaOne conference had several related announcements but I'll leave those to the marketing folks. The JCP already approved a few new JSRs that will help ease the complexity of development – such as JSRs 175, 181, 198, and 201. Now, four new JSRs related to J2EE technology have been added to the effort.

JSR 220 will define Enterprise JavaBeans version 3. The main scope is to reduce EJB's complexity from a developer point of view not only by making use of metadata annotations but also by providing utility classes, more programmatic defaults, simplification of stateless session beans, and much more. JSR 221, JDBC version 4, proposes to make use of new Java programming language features such as annotations and generics as well as provide sets of utility classes. JSR 222, the next version of the Java API for XML Data Binding, aims to complete the support for all W3C XML Schemas. This expert group is expected to work closely with the new JSR 224, JAX-RPC 2.0, expert group to establish better support for several XML Schema data types. JSR 222 also expects to define the mapping of Java to XML Schema (version 1 is already defined the other way). And, last, JSR 223, version 2 of JAX-RPC – this effort proposes to add support for the latest W3C and WS-I standards such as SOAP 1.2 and WSDL 1.2. This JSR will work closely with a number of others such as JSR 181 (Web Services Metadata) and JSR 173 (Streaming API for XML).

A Scripting JSR

JSR 223, Scripting Pages in Java Web applications, introduces the basic technical foundation to bridge the scripting community and the Java community. This addresses a need for application developers who use technologies like PHP, ECMAScript, and Active Server Pages by setting a standard mechanism with which you can access Java objects. These Java objects could be in a Java servlet container or in a Java VM. While PHP will be the expert group's first focus to provide a binding for, other scripting languages are considered as well. The Java Servlet Specification (JSR 154) defines abstractions for Web application context, request, response, and so on. When writing Web applications, Java classes and objects are developed that interact in well-defined security, resource, and class loader contexts. The JSR describes how this is exposed to scripting languages.

Fifteen ME EC Members?

Those of you who watch the ballot outcomes closely may have noticed that there are 15 voting ME EC members while there are 16 for SE/EE EC. Zucotto Wireless, which was elected to the ME EC in 2000 and reelected in 2001, has unfortunately closed its doors. Its seat will be vacant until the elections in November this year. Via this forum I would like to thank the folks of Zucotto for their hard work and contributions to the Java community and wish them well in their new adventures.

That's it for this month. I am 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

JAVA DEVELOPER'S JOURNAL Advertiser Index

Advertiser	URL	Phone	Page
Asperon Corporation	www.asperon.com		34
Borland Software Corporation	go.borland.com/j1		4
CTIA Wireless I.T.	www.ctiashow.com		59
Empirix Inc.	www.empirix.com/know	866-228-3781	7
ESRI	www.esri.com/mapobjectsjava	888-332-2320	21
Extentech	www.extentech.com/jdjsale/		35
Fair, Isaac & Company	www.fairisaac.com/rules		Cover III
InetSoft Technology Corp.	www.inetsoft.com/jdj	888-216-2353	33
Infragistics, Inc.	www.infragistics.com	800-231-8588	14-15
InstallShield Software Corp.	www.installshield.com/mpjdj		23
iSavix	http://isavix.net	703-689-3190	55
JetBrains	www.intellij.com		9
Kenetiks, Inc	www.kenetiks.com	888-KENETIKS	29
LinuxWorld Conference & Expo	www.linuxworldexpo.com		56-57
LinuxWorld Magazine	www.sys-con.com	888-303-5282	63
New Atlanta Communications	www.newatlanta.com		49
Northwoods Software Corp.	www.nwoods.com/go	800-434-9820	64
Oak Grove Systems	www.oakgrovesystems.com/jdj	818-880-8769	39
Oracle	oracle.com/experts	800-633-1072	11
Parasoft Corporation	www.parasoft.com/jdj2	888-305-0041	19
QUALCOMM Incorporated	www.qualcomm.com/brew		47
Quest Software, Inc.	http://java.quest.com/performasure/jdj		13
Quest Software, Inc.	http://java.quest.com/qcj/jdj		53
Quest Software, Inc.	http://java.quest.com/jclass/jdj		Cover IV
RealObjects	www.realobjects.com		25
RefactorIT	www.refactorit.com		60
ReportingEngines	www.reportingengines.com/info/JDJ_July_ere.jsp	888-884-8665	17
ReportingEngines	www.reportingengines.com/info/JDJ_July_ese.jsp	888-884-8665	26-27
ReportMill Software	www.reportmill.com/webstart	214-513-1636	43
Software FX	www.softwarefx.com		31
Sonic Software	www.sonicsoftware.com		Cover II
Sybase TechWave 2003	www.sybase.com/techwave2003		61
WebAppCabaret	www.webappcabaret.com/jdj.jsp		41
Web Services Edge West 2003	www.sys-con.com	201-802-3069	66-67
Wily Technology	www.wilytech.com	888-GET-WILY	51
Zero G	www.zerog.com	415-512-7771	3

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.

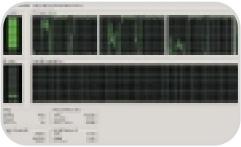
Next Month

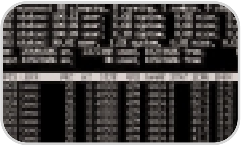
Understanding the Java Classloading Mechanism

The Java platform was designed to be a robust, secure, and extensible platform supporting the mobility of code and data. The Java classloader is a key component in the Java Virtual Machine (JVM) toward the realization of these goals. This article presents the Java classloader architecture and the implications of classloaders on platform security and extensibility.

Hyper-Threading Java

In early 2002 Intel became the first chip manufacturer to release a processor incorporating a new technology known as Simultaneous Multi-Threading or SMT. This article will explain the concepts of SMT in layman's terms, present the development of an *N*-thread benchmarking suite, and use that suite to produce concrete results of multi-threaded benchmarks on HT and non-HT systems.



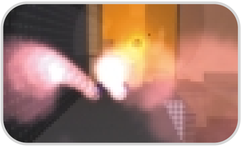


Java Games Development

I recently had a hankering to play an older (not ancient) PC game that I used to enjoy. I would even be tempted to dish out a few bucks to buy retro games, if I could only be sure that they were going to actually run successfully. How much easier would this process have been if some of these games had been written in Java? This had us thinking at *JDJ* (yet again) about games development in Java. Which is the reason we've gathered together an eclectic group of interested parties to discuss that very topic – Java Games Development

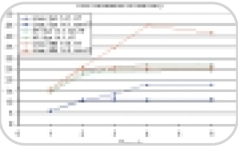
Practical Integration of Java-Oriented Technologies and 2.5/3G Platforms

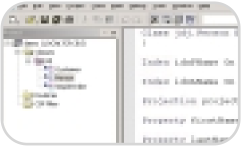
While deploying services through 3G and 2.5G wireless terminals presents challenges quite different from those encountered in PCs and servers, many familiar environments have been adapted to resource-constrained devices. This article discusses the practical integration of Java-oriented technologies and 2.5G/3G platforms and considerations you should understand when dealing with resource constraints and the management of software components through development, testing, deployment, and maintenance.



Caché 5 by InterSystems

One of the key challenges facing Java developers is that their object-oriented applications use data stored in relational databases. The result: time- and cost-intensive mapping between the two paradigms. InterSystems' Corporation develops and markets a post-relational database management system called Caché that's designed to address this challenge by eliminating the impedance mismatch between objects and tables.





The Lights Are On, but No One's Home

When Govindavajhala Sudhakar, a Princeton college student from Bangalore, presented a paper on JVM security at an IEEE symposium on computer security, the press naturally took notice. In addition, the ink is still wet on stories of how Muhammad Danka took only a few minutes to find a technique that allowed him to reset passwords of any Microsoft Passport user's account – www.siliconvalley.com/mld/siliconvalley/5822963.htm.

College students seem to be particularly adept at hacking and, with this in mind, CNET reported: "New hacking tool sees the light," http://news.com.com/2100-1009_3-1001406.html.

What Govindavajhala did was to create a Java applet with two classes, A and B (www.cs.princeton.edu/~sudhakar/papers/memerr.pdf). The program creates a single instance of A and fills the remainder of the heap with instances of B that point to the singleton A. A and B are defined so that the size they occupy in the JVM (including their object header) is a power of 2. If one of the bits in the JVM where B points to A were to flip, then there's a chance it would likely point to the base of one of the B objects instead.

It's hard to see how this corrupted heap is little more than an academic "so what?" However, the paper assures us that this can be used to read and write arbitrary JVM memory and thereby poses a security threat. Giving the benefit of the doubt on this issue, how's this bit flip going to occur on the PC where the attack applet is lurking?

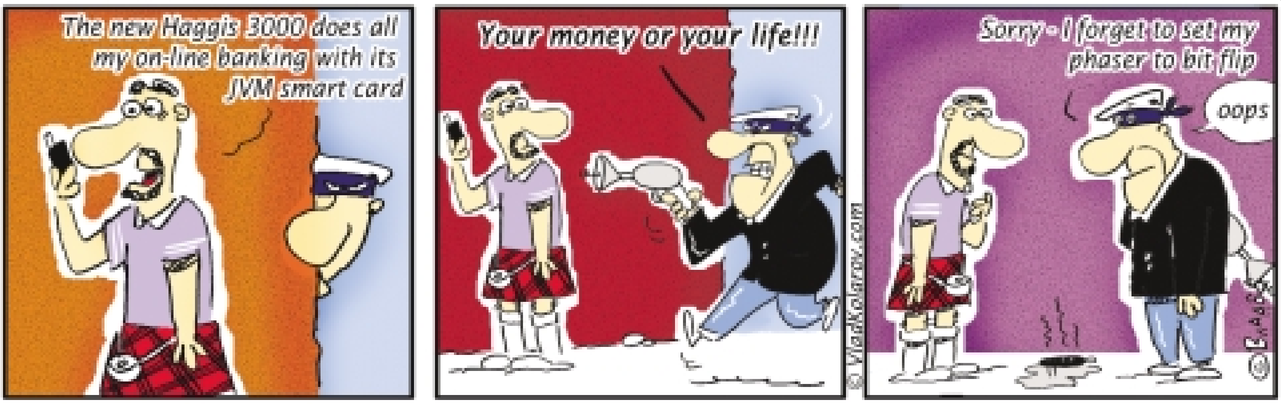
One method described is to rely on chance because when a cosmic ray interferes with the RAM holding the JVM, random bit flips can occur. Not content with waiting for this event to happen, the paper describes how to take apart a smoke detector to create a source of alpha particles, or use high-energy protons created by particle accelerators, although the favored technique is infrared radiation. The "researchers" opened the back of their PC and shone a 50-watt light bulb onto the memory chips. "As we were fine-tuning this experiment, we found that introducing large numbers of memory errors would often cause the operating system not only to crash, but to corrupt the disk-resident software so that reboot was impossible without reinstallation of the operating system." This is some kind of research euphemism for "Instead of our perfectly engineered single bit flip, we kept frying the hard drive by mistake."

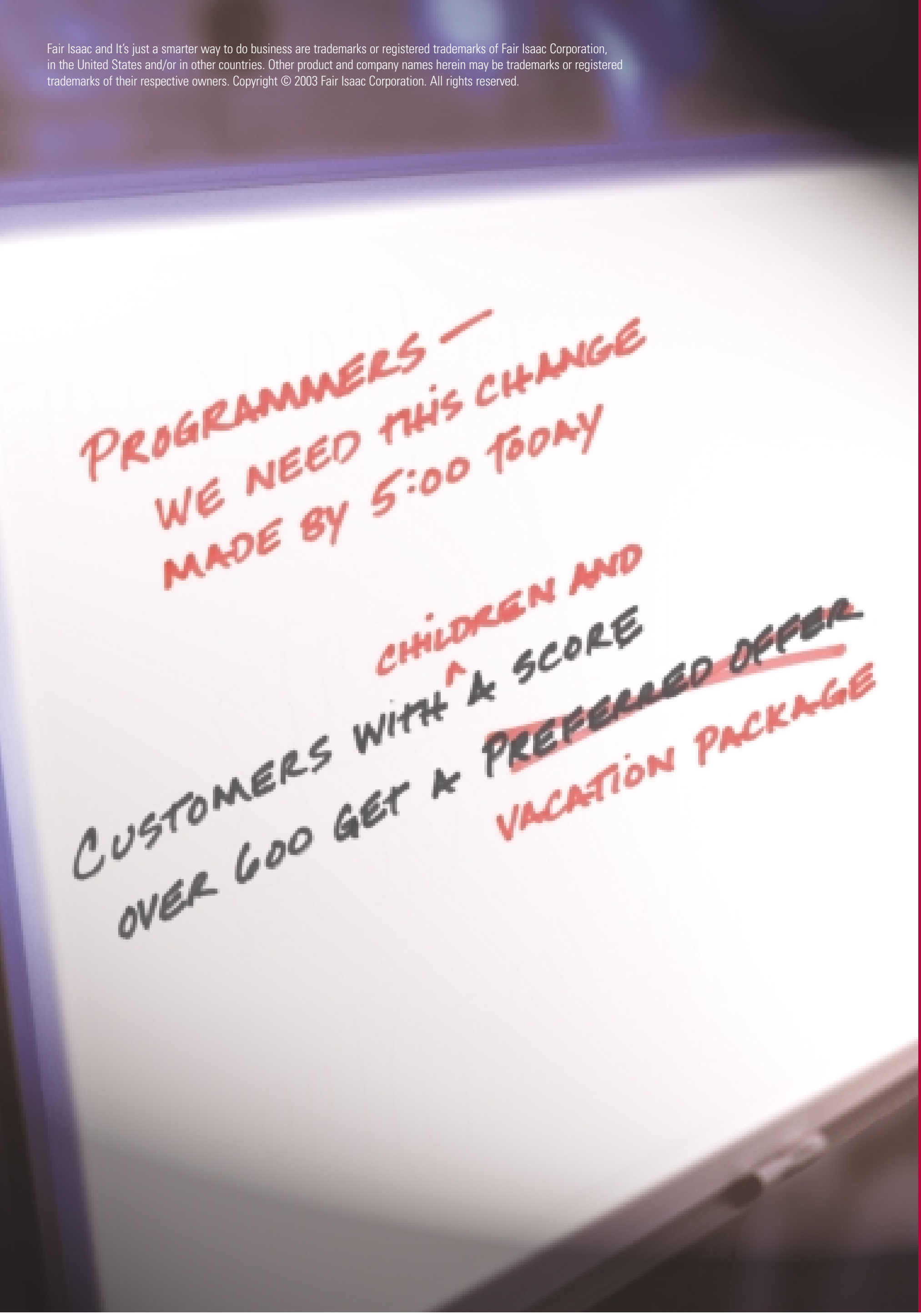
Not content with the unfortunate experimental side effect of frazzling their disk, the authors then go on to describe how a real attacker would not have the luxury of opening the box anyway, and how for a desktop PC "the attacker would have to heat the entire box in an oven." Remember readers, the attacker is trying to just flip a single bit in the JVM heap containing his string of B objects, and he's just put your PC into an oven. What happens if he overcooks it and your prized 3GHz Pentium comes out well done rather than rare? This is covered with the superb understatement, "We don't know whether the memory would become unreliable before other components failed," or in other words "If your PC gets turned to toast, it's all in the name of scientific progress."

When questioned by CIOL, www.ciol.com/content/developer/2003/103051401.asp, Govindavajhala stated, "Now, in India, some places go to 50ºC in the summer. Probably bits are already flipping in my homeland. Now, all I need to do to take over a good number of machines in India is to put this applet up on my Web page and wait for hits from India in summer. Computers of a billion people are at stake."

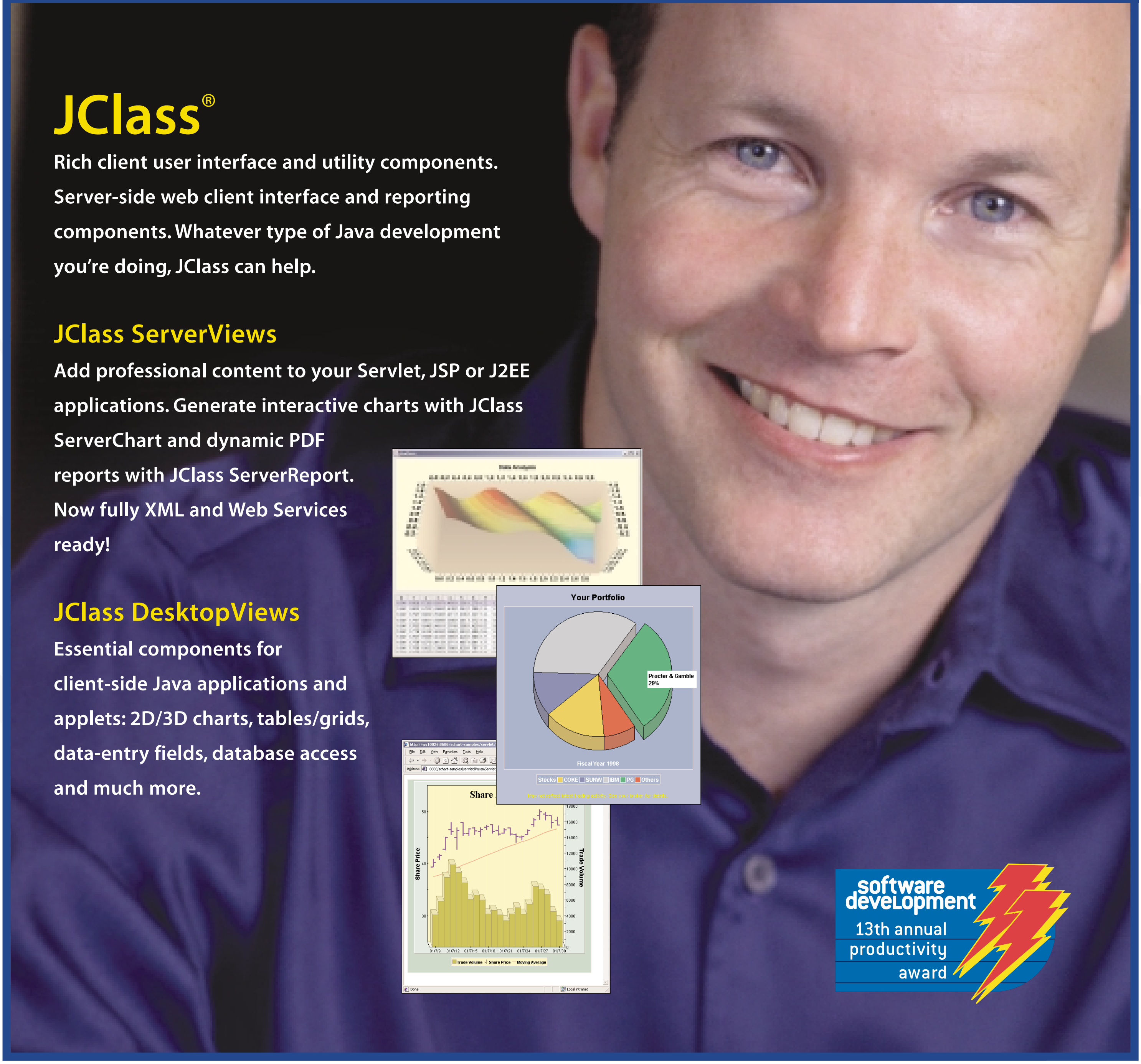
I think perhaps after spending too long in the sun himself, it's not only the PC's bits that have flipped. What's next for the IEEE, "Security Flaw: Monkeys with typewriters break 128-bit encryption."

Java Dudes


The new Haggis 3000 does all my on-line banking with its JVM smart card
Your money or your life!!!
© VladKolarov.com
Sorry - I forget to set my phaser to bit flip
oops


PROGRAMMERS —
WE NEED THIS CHANGE
MADE BY 5:00 TODAY
CHILDREN AND
CUSTOMERS WITH ^ A SCORE
OVER 600 GET A ~~PREFERRED OFFER~~
VACATION PACKAGE

The only Java components you need for J2EE or Swing development



JClass®

Rich client user interface and utility components. Server-side web client interface and reporting components. Whatever type of Java development you're doing, JClass can help.

JClass ServerViews

Add professional content to your Servlet, JSP or J2EE applications. Generate interactive charts with JClass ServerChart and dynamic PDF reports with JClass ServerReport. Now fully XML and Web Services ready!

JClass DesktopViews

Essential components for client-side Java applications and applets: 2D/3D charts, tables/grids, data-entry fields, database access and much more.

QUEST SOFTWARE®

Evaluate and experience JClass today - visit:

http://java.quest.com/jclass/jdj