JAVA™ DEVELOPER'S JOURNAL

The World's Leading Java Resource

August 2003 Volume:8 Issue:8

www.JavaDevelopersJournal.com

WE'VE ELIMINATED THE NEED FOR MONOLITHIC BROKERS. THE NEED FOR CENTRALIZED PROCESS HUBS. THE NEED FOR PROPRIETARY TOOL SETS.

Introducing the integration technology

YOU WANT.

Business Integration Suite

Introducing the Sonic *Business Integration Suite*. Built on the world's first enterprise service bus (ESB), a standards-based infrastructure that reliably and cost-effectively connects applications and orchestrates business processes across the extended enterprise. Extend your reach, cut costs and enhance your business agility. Thus eliminating one more thing – your headaches. To learn more, visit **www.sonicsoftware.com.**

Connect_Everything.
Achieve_Anything.™

Why do industry leaders choose Zero G?

Because, Zero G is the industry leader in multi-platform software deployment and updating. But, it didn't just happen overnight. Since 1996 we have listened, adapted, and built our award-winning InstallAnywhere® and PowerUpdate® products based on constant interaction with our customer-partners,That's why they have become the developers' deployment solution of choice worldwide. It's no wonder that industry leaders like Sun Microsystems, Novell and Borland choose us.

ZERO G
SOFTWARE

YOUR SOFTWARE DEPLOYMENT PARTNER

WWW.ZEROG.COM

BEA Users
Get Oracle For Free

Free Oracle9*i*AS Java licenses
For 1/2 your current BEA Java support fee
For details, call 1.800.633.0538

ORACLE®

For important conditions, limitations,
and restrictions call 1.800.633.0538

oracle.com/switch

JAVA DEVELOPER'S JOURNAL

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 Newsstand Distrubution
Curtis Circulation Company, New Milford, NJ

Alan Williamson
Editor-in-Chief

Everything's a $1

It has been a very busy month. SYS-CON Media recently began its complete overhaul of the LinuxWorld.com Web site, which we now publish as the online counterpart to ***LinuxWorld Magazine***, which we're launching this month at LinuxWorld in San Francisco. With this overhaul came all the logistics of moving and hosting the Web site – LinuxWorld.com having been previously run by International Data Group (IDG). Naturally we went with a Linux-based infrastructure as it would have been sacrilegious to do anything else!

This was your classic open source solution; no software licenses to pay and the download was quick and painless. The term "open source" doesn't sit too well with me though; I think it needs to be called what it is. I don't feel comfortable using it – let me explain.

When I downloaded and installed Apache I didn't need the source code. When I ran up the server and configured it, I didn't need the source code. When I read the documentation on how to configure some of the more advanced options, again, I didn't need to refer to the source code.

For us and the vast number of sites running Apache (nearly 27 million according to NetCraft, July 2003), the fact that Apache's source code is available doesn't matter to us. We are neither skilled nor have the time to actually look at it properly. For us, it's free software. I could paint the same story using MySQL, JEdit, Eclipse, or Ant.

Are you choosing (insert your free product of choice) because it's free or because it's open source? I hope you're choosing it because it's good and the best tool for the job. But what if they decided to start charging for it tomorrow. Would you pay for it? Would a nominal charge be acceptable? For example, would $1 per installation be too much? For Apache that could be $27 million in revenue.

There is no such thing as free in legal terms. A contract always has to be worth at least something and the usual amount is $1. Why? Probably because it's not of any monetary significance for anyone to complain. I would be more than happy to pay the "nominal" amount and put back into the system for the original developers' efforts.

But the term "free software" or "freeware/shareware" doesn't seem to be a sexy enough title. It's been around for years but still has a stigma attached to it. It doesn't capture the hearts and minds of the masses like the term "open source" does. Free software seems to have a perceived notion that it's of inferior quality to its commercial counterparts. As we all know, nothing could be further from the truth. The list price for a particular piece of software in no way guarantees its quality.

Open source is too misleading. The romantic notion of finding a bug, fixing it yourself, applying it back into the main build for the benefit of the larger community is so far from the truth that it's laughable. Open source projects (the successful ones) are very closely controlled with only a handful of designated people allowed to commit changes. But this is one of the reasons they are successful; the software is kept under tight control and not allowed to morph out of hand.

But the term "free" doesn't work either. Moving back to the model of $1-per-download, we could give it another name and call it "nominal licensing." Needs to be catchier than that I fear! But it would lift us away from the "freeware" stigma and be friendlier than open source.

We are moving toward a freer choice of tools/software, allowing us to choose based on functionality and performance as opposed to which one has the cheapest price tag.

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

Unleashing Tiger

Kirk Pepperdine

Up till now, changes to Java have been pretty much constrained to APIs and the inner workings of the Java 2 platform. All of this will change once the JDK 1.5 has been released. The extent of these changes was revealed in a recent interview with Joshua Bloch (http://java.sun.com/features/2003/05/bloch_qa.html). As illuminating as the interview was, it left me with a number of questions, so I started to search for answers by reading JSR 201.

If you read this JSR, you'll find that it doesn't include a critical discussion of the proposed changes, nor is there any supporting documentation that includes a record of such a discussion. In fact, I was unable to find a public record of any critical discussion. Surely those discussions took place. What alternatives were proposed and why were they ultimately rejected? Do the proposed changes represent a direction for the evolution of the language? Take autoboxing, for instance. What problem is it trying to solve and is there an alternative that offers a better solution? Here are some thoughts on this point.

Autoboxing is a concise notation for working with instances of the immutable classes that are used to wrap primitive types. It will allow us to replace code such as "new Integer(((Integer)map.get(key)).getInt() + 1)" with "map.get(key) + 1". While this code is easier on the eye, does this new syntax tackle the fundamental issue that objects and primitive types are like water and oil? They don't mix unless you add an emulsifier. In Java, the emulsifying agents have been wrapper classes, one for each primitive type. As necessary as these wrapper classes are, they do hinder our ability to operate on the underlying primitive unless we resort to the unpleasant-looking code listed above. Instead of solving the problem, wrapper classes have seemingly shifted the problem elsewhere, which resulted in the JCP Expert Group recommending that autoboxing be added to the language. This begs the question: Couldn't we have solved the problem instead of passing it along? If mixing objects and primitives is the problem, it seems reasonable to suggest that we shouldn't mix them. Instead, let's promote primitives to the status of first class objects with all the rights and privileges. In doing so, we would eliminate the need for wrapper classes and, consequently, for autoboxing.

In the past when I suggested that primitives should be promoted, I was told that this would be incredibly inefficient. My counter to this argument focuses on two points. First, being a first class object is a syntactic or notational convenience. The final compiled representation of integers and other primitives could (and should) be the choice of the implementers (sound familiar?). The second point was really in response to the comments about the differences in efficiency between operating on primitive types and on objects. But what of these differences? Do we really have to experience the cost of method lookups for each operation? To this I reply: Java is a typed language. In other words, we are giving the compiler big hints as to what we are doing. The compiler should be able to use these hints to do the right thing. In fact, current compiler optimizations are the result of far more complicated evaluations. One last point: autoboxing is implicitly creating both new objects and garbage. Though this is not bad in itself, it does remind me of the performance problems that can result from using + to simplistically concatenate strings.

I first thought of offering a counterargument to my proposal, but then realized that it's not my role to do so; it's yours. Furthermore, shouldn't it be the role of the JCP to foster healthy debates on all of its proposals so that we, as a community, know and understand that we are getting the best choices possible? After all, we, the average Java developers, are the ones who sit in the trenches of the Java community, and the JCP can only be our process if we all become more involved in it. You can send your comments to the JSR 201 and JSR 215 Expert Groups.

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

kirk@javaperformancetuning.com

NEW!
Release 5.1
out now!

Java thin clients made easy

Ultra Light Client with full GUI capability

easy development

- no worries about client-server code split
- no technology mix: pure server-side Java development
- widgets transparently split between client and server
- extensible Swing-based widget set incl. source code

easy deployment and operation

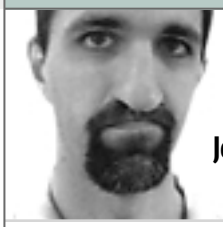
- application deployment on server only
- lean and generic display engine on client
- optimized for low-bandwidth communication
- seamless J2EE integration (Web/EJB container)
- Web integration (applet, Java Web Start)

canoo.com

Canoo Engineering AG
Kirschgartenstrasse 7
CH-4051 Basel
Switzerland

Tel +41 61 228 94 44
Fax +41 61 228 94 49
http://www.canoo.com/ulc/
ulc-info@canoo.com

Download your free trial today!



Joseph Ottinger
J2EE Editor

Chaos and Order

Chaos. Anthropomorphically speaking, it wants to go everywhere.

Order. It wants to be everywhere too, and is willing to fight chaos to do it.

Michael Moorcock used to write lots of fundamentally depressing books about this very idea, and you can see it everywhere today – politically speaking, in the U.S. You have order being imposed as much as is possible on a culture that tends to thrive on chaos; more relevant to this magazine, you have industry standards trying to make sure that everyone marches in straight lines, while rogue coders (so to speak!) spend time writing marvelous, new code underground.

Both sides are good; innovative solutions tend to come initially from the chaotic underground, and tested solutions tend to grow out of the orderly standards bodies taking the new solution and rigorously smacking it about until it's able to handle real load. The concept comes from beneath, and the validity of the concept nearly always comes from the office.

Unfortunately, both sides tend to view the other with suspicion. The ones who chant "standards! standards!" are corporate stooges, to the bazaar-based bunch; likewise, the madding crowd is viewed as a set of subversive, rogue elements by those wearing ties. Neither group trusts the other, and often both groups insist that the opposition isn't necessary.

It's important for us to remember that chaos exists within the framework of order, or else we can't recognize what it is – likewise, order stifles creativity and we need creativity right now.

This can be applied by using software and participating in the process. For us, that means that people who understand the standards would benefit from using open software and modifying it to comply with what the industry requires. The open software authors likewise need to understand that standards exist for a reason, that there's a benefit to them. That's already somewhat in place – remember when there used to be shouting matches over where to place brackets? You still have that every now and then, but there's a common standard that's dominating now, so the fights over code format are sporadic and short. That sort of scenario can play itself out in how code works, too, just like it has played out in how code is formatted.

Of course, if everyone decides to only play by the rules, then creativity is stifled. I'd rather this not happen, obviously. It's important to see new ways of doing things, even if only so they can be rejected. Consider object databases, which are excellent solutions for some vertical applications; when your domain is limited to those applications and your assumptions can be applied, they're great! (Look at Prevayler and its claims of incredible speed.) However, the assumptions can be nasty; object databases can be crippling if you need to process the data from a language that doesn't represent objects the same way. (Consider a COBOL program trying to modify a Prevayler dataset. It can be done, to be sure…but let's just say that most people would rather gnaw off their own neck first.)

Yet, it's still important to have these tools. You'll certainly find solutions that are vertical enough to use these products well and provide a reason for their existence. Who knows, maybe their authors or users will find a way to bridge the conceptual divide and make the products more generically useful and take over, but the "generically useful" aspect will almost always come from larger, organized bodies imposing a structure over what the products do and how.

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

josephottinger@sys-con.com

Understanding the Java Classloading Mechanism

Extend the Java platform

Rohit Chaudhri

The Java platform was designed to be robust, secure, and extensible in order to support the mobility of code and data. The Java ClassLoader in the Java Virtual Machine (JVM) is a key component in the realization of these goals.

The JVM is responsible for loading and executing code on the Java platform. It uses a ClassLoader to load Java classes into the Java runtime environment. ClassLoaders are architected so that at start-up the JVM doesn't need to know anything about the classes that will be loaded at runtime. Almost all Java-based containers such as EJB or servlet containers implement custom ClassLoaders to support features like hot deployment and runtime platform extensibility. An in-depth understanding of ClassLoaders is important for developers when implementing such Java-based containers.

For enterprise developers who develop components that are deployed on these containers, this knowledge will help you understand how the container works and with debugging problems. This article presents the Java ClassLoader architecture and discusses the implications of ClassLoaders on platform security and extensibility as well as a method to implement user-defined ClassLoaders.

The smallest unit of execution that gets loaded by a ClassLoader is the Java class file. A class file contains the binary representation of a Java class, which has the executable bytecodes and references to other classes used by that class, including references to classes in the Java API. Stated simply, a ClassLoader locates the bytecodes for a Java class that needs to be loaded, reads the bytecodes, and creates an instance of the java.lang.Class class. This makes the class available to the JVM for execution. Initially when a JVM starts up, nothing is loaded into it. The class file of the program being executed is loaded first and then other classes and interfaces are loaded as they get referenced in the bytecode being executed. The JVM thus exhibits lazy loading, i.e., loading classes only when required, so at start-up the JVM doesn't need to know the classes that would get loaded during runtime. Lazy loading plays a key role in providing dynamic extensibility to the Java platform. The Java runtime can be customized in interesting ways by implementing a custom ClassLoader in a program, as I'll discuss later.

Java 2 ClassLoader Delegation Model

Multiple instances of ClassLoaders exist in the Java 2 runtime, each loading classes from different code repositories. For instance, Java core API classes are loaded by the bootstrap (or primordial) ClassLoader. Application-specific classes are loaded by the system (or application) ClassLoader. In addition, an application can define its own ClassLoaders to load code from custom repositories. Java 2 defines a parent-child relationship between ClassLoaders. Each ClassLoader except the bootstrap ClassLoader has a parent ClassLoader, conceptually forming a treelike structure of ClassLoaders. The bootstrap ClassLoader is the root of this tree and thus doesn't have a parent. The relationship is depicted in Figure 1.

The following is a high-level classloading algorithm executed by a ClassLoader when a client requests it to load a class:

1. A check is performed to see if the requested class has already been loaded by the current ClassLoader. If so, the loaded class is returned and the request is completed. The JVM caches all the classes that are loaded by a ClassLoader. A class that has previously been loaded by a ClassLoader is not loaded again.
2. If the class is not already loaded, the request is delegated to the parent ClassLoader, before the current ClassLoader tries to load it. This delegation can go all the way up to the bootstrap ClassLoader, after which no further delegation is possible.
3. If a parent fails to return a class because it was unable to load it, the current ClassLoader will then try to search for the requested class. Each ClassLoader has defined locations where it searches for classes to load. For instance, the bootstrap ClassLoader searches in the locations (directories and zip/jar files) specified in the sun.boot.class.path system property. The system ClassLoader searches for classes in the locations specified by the classpath (set as the java.class.path system property) command-line variable passed in when a JVM starts executing. If the class is found, it's loaded into the system and returned, completing the request.
4. If the class is not found, a java.lang.ClassNotFoundException is thrown.

Implementing a Java 2 Custom ClassLoader

As mentioned earlier the Java platform allows an application programmer to customize the classloading behavior by implementing a custom ClassLoader. This section shows how.

ClassLoaders (all but the bootstrap ClassLoader, which is implemented in native code in the JVM) are implemented by extending the java.lang.Class Loader class. The following code shows the relevant methods of the Java 2 ClassLoader API:

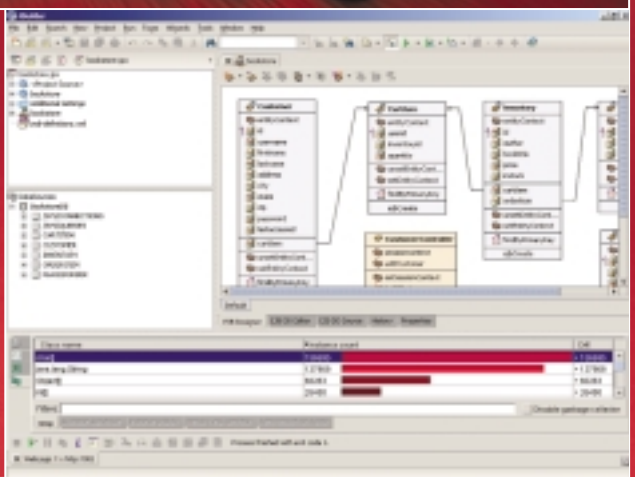
```
1.  public abstract class ClassLoader extends Object {
2.  protected ClassLoader(ClassLoader parent);
3.  protected final Class defineClass(
4. String name,byte[] b,int off,int len)
5. throws ClassFormatError
5. protected Class findClass(String
7. className) throws ClassNotFoundException
6. public class loadClass(String className)
7. throws ClassNotFoundException
8.}
```

Each ClassLoader is assigned a parent when it's created, as per the parent-delegation model. Clients invoke the loadClass method on an instance of a ClassLoader to load a class. This initiates the classloading algorithm as

Borland® JBuilder®

Ten things you didn't know about the #1 IDE for Java™

1) Deep integration with BEA, Borland, IBM, Oracle, SunOne, Sybase, and Tomcat application servers
2) JBuilder® Developer Edition for hard-core developers
3) Award-winning Borland® Optimizeit™ Suite now included
4) Import project source from any IDE or editor
5) State-of-the-art refactoring capabilities
6) Fully integrated Apache™ ANT build capabilities and JUnit testing
7) Advanced flexible code formatting and Borland® ErrorInsight™
8) New developers quickly productive with advanced tools and wizards
9) Dynamic web support using struts, servlets, JavaServer Pages, XML, and Web Services
10) Standards compliance to deploy your code anywhere



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.

Visit **go.borland.com/j6**

Learn from the experts at the premier event for technical education — the 2003 Borland Conference. Learn more: www.borland.com/conf2003s

Borland®
Excellence Endures™

DEFINE	DESIGN	DEVELOP	TEST	DEPLOY	MANAGE
Borland® CaliberRM™	Borland® Together® ControlCenter®	Borland JBuilder	Borland Optimizeit Suite	Borland® Enterprise Server, AppServer™ Edition	Borland® StarTeam®

explained earlier. Prior to Java 2, the loadClass method in the java.lang.ClassLoader class was declared abstract, requiring custom ClassLoaders to implement it when extending the java.lang.ClassLoader class. Implementing the loadClass method is rather complicated, so this has been changed in Java 2. With the introduction of the ClassLoader parent delegation model, java.lang.Class Loader has an implementation of the loadClass method, which is essentially a template method that executes the classloading algorithm. The loadClass method invokes the findClass method (introduced in Java 2) in Step 3 of the classloading algorithm. Custom ClassLoaders should override this method to provide a custom way of locating and loading a Java class. This greatly simplifies the implementation of a custom ClassLoader. Listing 1 shows some code from the CustomClassLoader.java class (the complete source code can be downloaded from www.sys-con.com/java/sourcec.cfm), which loads classes from a repository specified in the constructor.

The findClass method invokes loadFromCustomRepository that searches for the given class in the repository and, if found, reads and returns the bytecodes for the class. The raw bytecodes for the class are passed into the defineClass method implemented in the java.lang.Class Loader class, which returns an instance of the java.lang.Class object. This makes a new class available to a running Java program. The defineClass method also ensures that a custom ClassLoader does not redefine core Java API classes by loading them from a custom repository. A SecurityException is thrown if the class name passed to defineClass begins with "java".

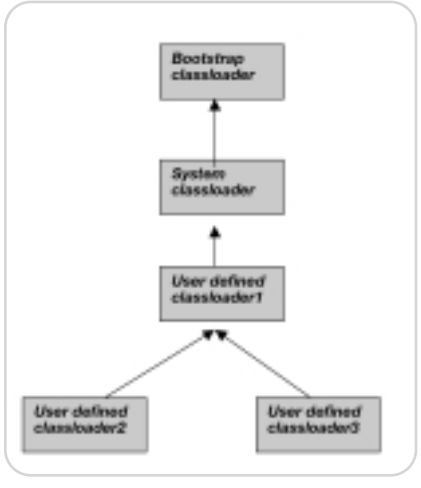


Figure 1 The parent delegation model

It should be noted that at start-up, the JVM doesn't need to know anything about the class represented by the string passed into the loadClass method. A subsequent section shows how a program can use the CustomClassLoader.

Rohit Chaudhri is a member of the mobile platforms and services lab at Motorola. The team investigates end-to-end application platforms for the wireless Internet. He has an MS in computer science and is a Sun Certified Programmer for the Java 2 Platform.

rohit.chaudhri@motorola.com

Deviations from the Java 2 Delegation Model

The Java 2 delegation model cannot be followed in all situations. There are cases in which ClassLoaders have to diverge from the Java 2 model. For instance, the servlet specification recommends (section 9.7) that a Web application ClassLoader be implemented so that classes and resources packaged in the Web application archive are loaded in preference to classes and resources residing in container-wide JAR files. To meet this recommendation, a Web application ClassLoader should search for classes and resources in its local repository first, before delegating to a parent ClassLoader, thus deviating from the Java 2 delegation model. This recommendation makes it possible for Web applications to use different versions of classes/resources than those being used by a servlet container. For example, a Web application might be implemented using features available in a newer version of an XML parser than the one being used by a servlet container.

A Web application ClassLoader that meets the recommendation of the servlet specifications can be implemented by overriding the loadClass method of the java.lang.Classloader class. The loadClass method of such a custom ClassLoader may look similar to Listing 2.

Applications of ClassLoaders

ClassLoaders provide some powerful features that can be utilized in Java programs. This section discusses a few ways in which they can be used.

Hot Deployment

Upgrading software in a running application without restarting it is known as hot deployment. For a Java application, hot deployment means upgrading Java classes at runtime. ClassLoaders play an important role in Java-based application servers to achieve hot deployment. This feature is exploited in most Java-based application servers like EJB servers and servlet containers. A ClassLoader cannot reload a class that it has already loaded, but using a new instance of a ClassLoader will reload a class into a running program. The following code from TestCustomLoader.java illustrates how hot deployment may be achieved in a Java application:

```
1. ClassLoader customLoader = new
2. CustomClassLoader(repository);
3.    loadAndInvoke(customLoader,classToLoad);
4.    System.out.println("waiting.Hit
5.    Enter to continue");
6.    System.in.read();
7. customLoader = new CustomClassLoader
8. (repository);
9.    loadAndInvoke(customLoader,classToLoad);
```

An instance of the CustomClass Loader is created to load classes from a repository specified as a command-line parameter. loadAndInvoke loads a class, HelloWorld, also specified as a command-line parameter, and invokes a method on its instance, which prints a message on the console. While the program is waiting for user input at line 6, the HelloWorld class can be changed (by changing the message that gets printed on the console) and recompiled. When the program continues execution, a new instance of CustomClassLoader is created at line 7. When loadAndInvoke executes line 9, it loads the updated version of HelloWorld and a new message is printed on the console.

Modifying Class Files

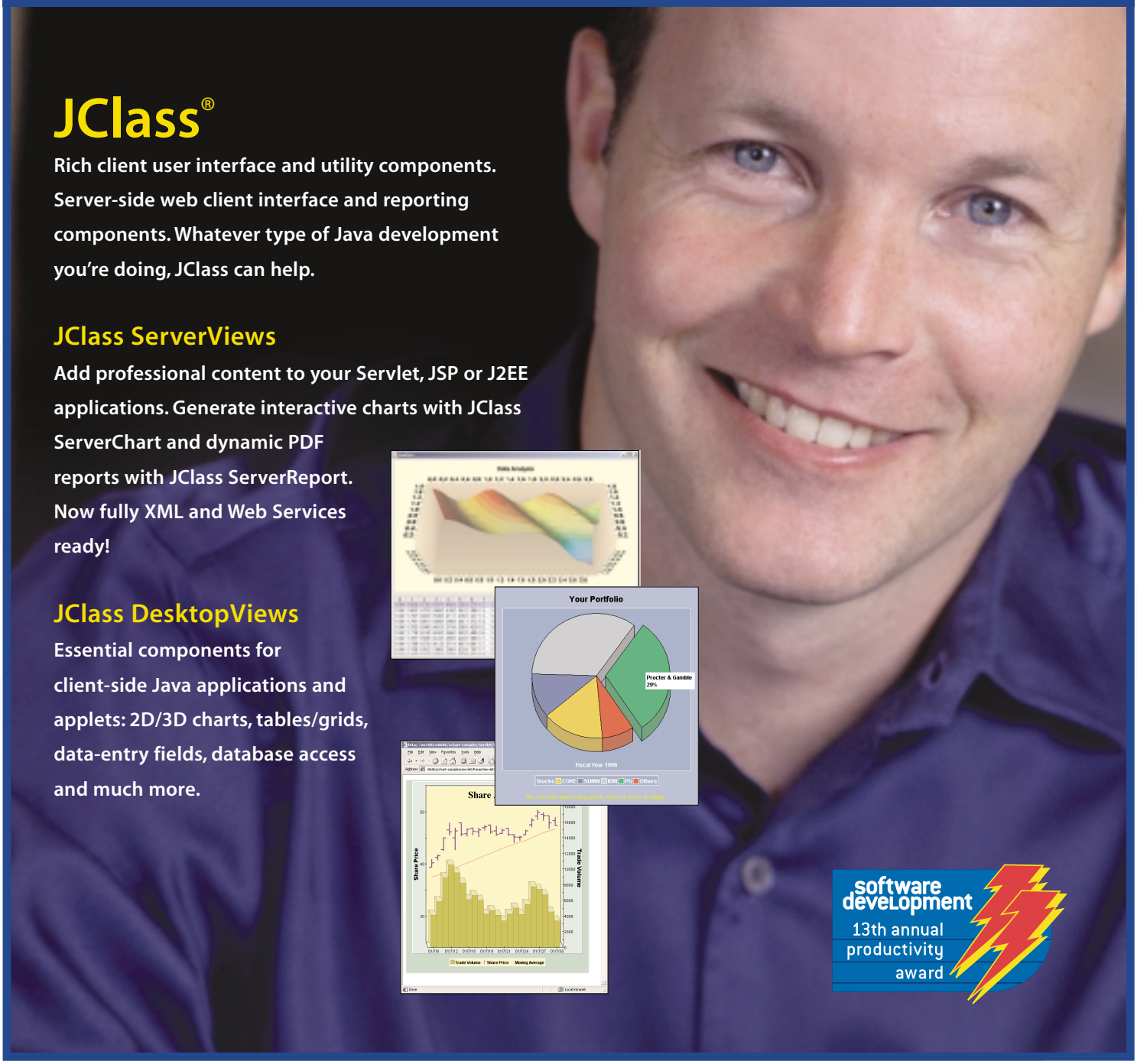
A ClassLoader searches for bytecodes of a class file in the findClass method. After the bytecodes have been located and read into the program, they may be modified before invoking defineClass. For example, extra debugging information may be added to the class file before invoking defineClass. Class file data for some secure applications may be stored encrypted in a repository; the findClass method can decrypt the data before invoking defineClass. A program can generate the bytecodes on the fly instead of retrieving them from a repository. This forms the basis of JSP technology.

ClassLoaders and Security

Since a ClassLoader is responsible for bringing code into the JVM, it's architected so that the security of the platform may not be compromised. Each ClassLoader defines a separate namespace for the classes loaded by it, so at runtime a class is uniquely identified by its package name and the ClassLoader that loaded it. A class is not visible outside its namespace; at runtime there's a protective shield between classes existing in separate namespaces. The parent delegation model makes it possible for a ClassLoader to request classes loaded by its parent, thus a ClassLoader doesn't need to load all classes required by it.

The various ClassLoaders that exist in a Java runtime have different repositories from which they load code. The idea behind separating repository loca-

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.

software development 13th annual productivity award

QUEST SOFTWARE®

Evaluate and experience JClass today - visit:

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

tions is that different trust levels can be assigned to the repositories. The Java runtime libraries loaded by the bootstrap ClassLoader have the highest level of trust in the JVM. The repositories for user-defined ClassLoaders have lower levels of trust. Furthermore, ClassLoaders can assign each loaded class into a protection domain that defines the permissions assigned to the code as it executes. To define permissions on code based on the system security policy (an instance of java.security.Policy), a custom ClassLoader should extend the java.security.SecureClassLoader class and invoke its defineClass method that takes a java.security.CodeSource object as a parameter. The defineClass method of SecureClassLoader gets the permissions associated with the CodeSource from the system policy and defines a java.security.Protection Domain based on that. A detailed discussion of the security model is beyond the scope of this article. Further details can be obtained from the book *Inside the Java Virtual Machine* by Bill Venners.

Summary

ClassLoaders offer a powerful mechanism by which the Java platform can be extended in interesting ways at runtime. Custom ClassLoaders can be used to achieve functionality not normally available to a running Java program. Some of these applications have been discussed in this article. ClassLoaders play an important role in some of the technologies offered by current J2EE platforms. For further details about the Java classloading mechanism, read *Inside the Java Virtual Machine*.

References

- Liang, S., and Barcha, G. "Dynamic Class Loading in the Java Virtual Machine": www.java.sun.com/people/gbracha/classloaders.ps
- Venners, B. (2000). *Inside the Java Virtual Machine*. McGraw-Hill Osborne Media.
- Neward, T. (2000). *Server-Based Java Programming*. Manning Publications Company.
- Gong, Li. "Secure Java Class Loading" http://java.sun.com/people/gong/papers/ieeeic98.pdf
- *The Java Servlet Specification:* http://jcp.org/aboutJava/communityprocess/first/jsr154/

Listing 1

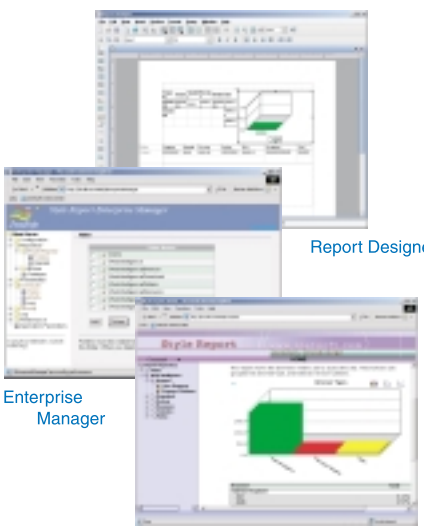
```
1. public class CustomClassLoader
2. extends ClassLoader {
3. //search repository
   4. private List classRepository;

5. public CustomClassLoader
6. (ClassLoader parent,String searchPath)
7. {…}

8. protected Class findClass(String

9. className) throws
10. ClassNotFoundException {
11. byte[] classBytes =
12. loadFromCustomRepository(className);
13. if(classBytes != null) {
14. return defineClass
15. (className,classBytes,0,classBytes.length);
16.    }
17. //else
18. throw new ClassNotFoundException(className);
19. }
20. }
```

Listing 2

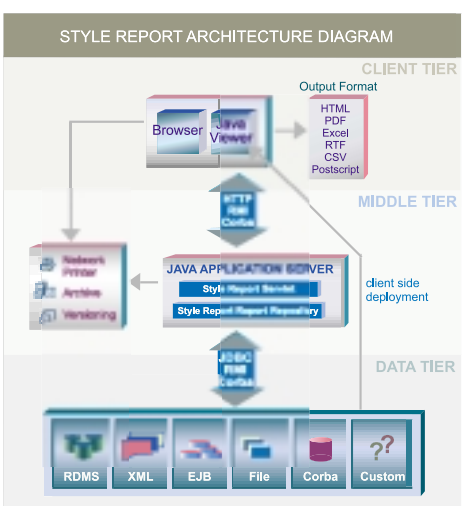
```
1. public Class loadClass(String name)
2. throws ClassNotFoundException
3. {
4.    //check if the class is already loaded
5.    Class loadedClass = findLoadedClass(name);
6.    if (loadedClass == null) {
7.    //search for class in local
          8.    //repository before delegating
9.    .....
10. //if class not found delegate to parent
11.  loadedClass =
13. this.getClass().getClassLoader().loadClass(name);
13. }
14. return loadedClass;
15. }
```

Java & Web Reporting:
with Java flexibility
without Java complexity

Style Report:
Information Vantage Point



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



Style Report Features

Flexible Visual Designer with JavaScript or VB Script	☑
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)

FORM-BASED AUTHENTICATION

EXTENDING J2EE

ActiveAuthentication

BY THOMAS BECK

When choosing among the different types of authentication mechanisms offered by J2EE Web containers, form-based authentication is almost always selected ahead of its alternatives: HTTP basic authentication and HTTPS client authentication. However, beneath the customizable user interface, form-based authentication presents several challenges to architects looking for a robust enterprise authentication solution.

These challenges often manifest themselves as login-page access errors that arise when applications look to extend the concept of protected resources upon which form-based authentication is predicated. This can occur when a system exhibits requirements that are considered fairly common for an enterprise application, such as:

- Authentication must be possible from the default form–based login page without trying to first access a protected resource.
- Capture of authentication credentials must occur on multiple separate pages (e.g., via a login form that is a part of each nonprotected page).

In both cases, form-based authentication alone will not be able to satisfy the system requirements.

Available Options

There are three solutions that are most frequently recommended by application architects looking to satisfy either of the aforementioned requirements. The first and most commonly suggested one is to build a custom, servlet-based authentication mechanism. This solution, although robust and well documented, fails to leverage the infrastructure provided by container-managed security and requires that the authentication mechanism be implemented programmatically. The second solution is to subclass or interface directly with the authentication APIs provided by the Web container – such as Tomcat's AuthenticatorBase class. Such a solution can only be recommended to seasoned Java programmers and introduces a dependency on a particular Web container's internal APIs.

The third solution – and the topic of this article – is the extension of the existing J2EE form-based authentication mechanism to satisfy these requirements. This solution will enable an application to overcome some of the most commonly encountered form-based authentication hurdles without sacrificing the utility of the Web container's preexisting authentication capabilities or introducing Web container dependencies.

Form-Based Authentication

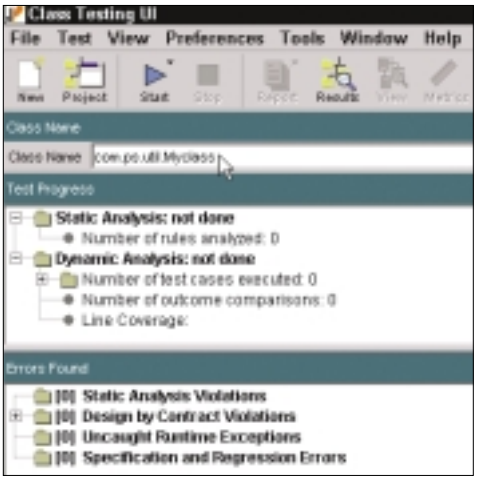
Before launching into a discussion on extending form-based authentication, it's important to first understand what form-based authentication is and is not. Please note that although a brief description of form-based authentication is provided here, more detailed descriptions and set-up instructions can be found in the references section at the end of this article.

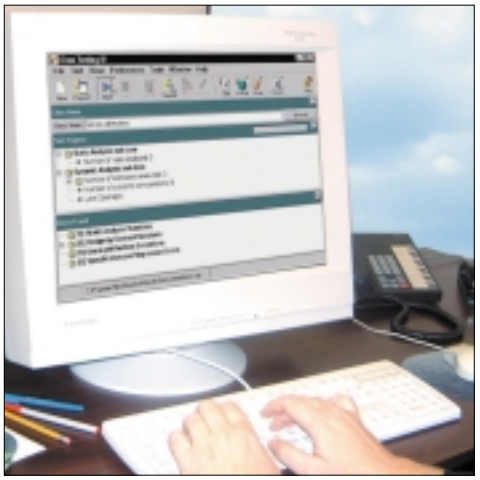
Form-based authentication is, at its core, a Java-specific, container-implemented authentication mechanism that allows the look and feel of the login screen to be customized. The login is performed via a form that must contain two fields for entering a username and a password, j_username and j_password, respectively, and a special container-recognized action – j_security_check.

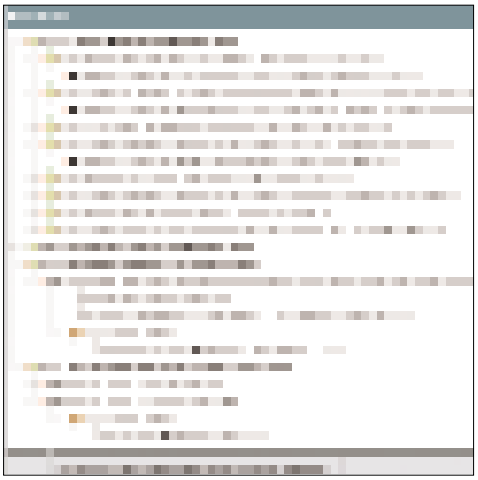
Beyond the login form, the application-specific implementation of form-based authentication is dependent on

PARASOFT
We make software work.™

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







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

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

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

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

●----- **For Downloads go to www.parasoft.com/jdj08. 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®

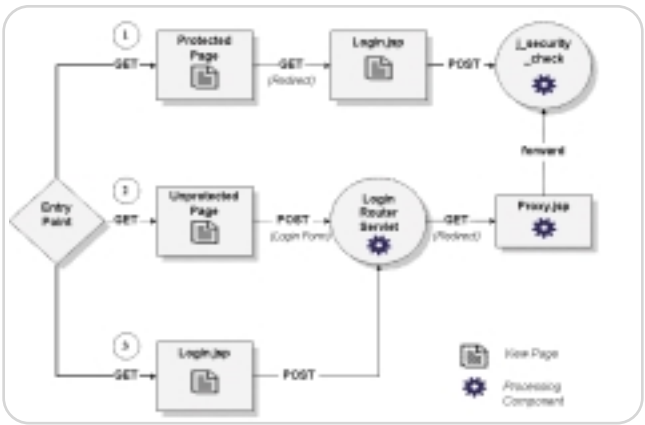


Figure 1 Active authentication flowchart

two very important elements in the Web application deployment descriptor: the login-config and security-constraint elements. The login-config element is used to indicate that an application is using form-based authentication and to specify the locations of the login and error pages to be used. The security-constraint element is used to define which resources are protected and to associate role-based constraints with these protected resources.

Although the implementation of form-based authentication is vendor-specific, the functional specifications are outlined in the Java servlet specification. When a user attempts to access a protected resource, the container checks the user's authentication. If the user is already authenticated and possesses a role that is authorized to access the resource (as defined in the security-constraint element), the requested resource is activated and a reference to it is returned. If the user is not authenticated, the following series of events occurs:

1. The login form is sent to the client and the original client request is stored by the container.
2. The client posts the login form back to the server, which then attempts to authenticate the client.
3. If authentication is successful, the user's roles are compared to the roles necessary to access the resource. If the user's roles authorize access to the resource, the client is redirected to the resource using the original request URL path stored by the container in the first step.
4. If authentication is not successful, the user is redirected to the error page defined in the login-config element.

The good news about form-based authentication is that the container takes care of a lot of the hard work. Authentication, authorization, redirection to the login page and back to either the protected resource or the predefined error page are all handled by the container. However, this level of container control over authentication and authorization carries with it several disadvantages that directly affect the ability of an application that uses form-based authentication to satisfy advanced authentication requirements.

The first disadvantage of container control is that the container maintains exclusive rights to the original client URL request for a protected resource. Generally, this parameter is neither accessible nor settable in a programmatic manner. The second disadvantage of container management of security is that, in most cases, it precludes the use of custom security filters. As an example, filters on the j_security_check action prove to be very unreliable across different application servers. The third and perhaps most important disadvantage is that container-controlled security does not support authentication unless the client is explicitly attempting to access a protected resource. This feature is often referred to as *lazy authentication*. That is, authentication is performed only when it is needed – not when your application would like it to be performed.

Active Authentication

Many systems that use form-based authentication also wish to perform authentication before it's absolutely required. Such is the case with the requirements outlined at the beginning of this article, including a login form on non-protected pages or a direct login from the login screen before accessing a protected resource – both of which will require "preemptive" authentication. A very compelling case could be made for such functionality if an application contained very few custom pages or wanted to customize its unprotected pages for particular user roles. In such cases, the application needs to actively engage in authentication when these services are not explicitly provided by form-based authentication. Note that this is different from a custom security solution in which the application assumes all authentication responsibilities. Active authentication, in this sense, implies that the application augments the form-based authentication services already provided by the container.

The first step in designing and implementing an active authentication system is to understand that there are a finite number of authentication entry points into a system. For each of these entry points, form-based authentication will either handle authentication for you or it will default to its lazy authentication posture and not handle authentication. For demonstration purposes, the remainder of this article will deal with a system that has three possible authentication entry points:

1. Direct access to protected pages
2. Active authentication from all unprotected pages via a login form included in each of the pages
3. Direct login from the login page defined in the Web application deployment descriptor

Figure 1 illustrates the active authentication flow for the model's three entry points. Please note that Figure 1 does not

"Many systems that use form-based authentication also wish to perform authentication before it's absolutely required"

include post authentication activities since these activities are controlled entirely by the container.

Of the three authentication entry points in our model, only point 1, direct access to protected pages, is handled by form-based authentication. Authentication entry points 2 and 3 will have to be handled explicitly by the application. You can easily test this assertion by entering valid credentials into a simple form with an action of j_security_check before you have tried accessing any protected resources. Doing this will result in an error such as "Invalid direct reference to form login page." In other words, the application will have to take care of referencing the form login pages to prevent this type of error from occurring.

For authentication entry point 2, which covers all unprotected pages, a special login include needs to be created with an action that posts the user's credentials to the Login Router Servlet. The Login Router Servlet sets the appropriate login session variables, makes note of the unprotected page that requested active authentication, and redirects to the Proxy JSP. The Proxy is defined in the Web application deployment descriptor as a protected resource. Therefore, any request for the Proxy will be subject to authentication by the container. The Proxy's job is to keep track of the original page that requested authentication. The Proxy then redirects back to the original page when the container has returned control to the Proxy.

Authentication entry point 3, direct entry from the default login page, requires that the page be smart enough to recognize the difference between a request to log in to a protected resource and an active authentication request. When active authentication requests are received, the login page should display some sort of selection mechanism that allows the user to choose a page to be routed to after authentication is complete. This option will not be required in the case of an attempt to access a protected resource.

"The first step in designing and implementing an active authentication system is to understand that there are a finite number of authentication entry points into a system"

By day, **Thomas Beck** *is actively involved in the design, development, and integration of enterprise systems as a senior consultant for Deloitte Consulting. By night, he is working on a book about developing enterprise software solutions with open source tools. Thomas holds Java certifications from both Sun and IBM.*

thomas_beck@acm.org

Authentication Synthesis

Now it should be apparent that although form-based authentication alone does not satisfy all the requirements of an enterprise application, it does provide a solid foundation upon which to build. The remainder of this article will focus on the four software objects that are needed to support the synthesis of active and form-based authentication. These four objects will be described below. Source code for these objects can be downloaded from www.sys-con.com/java/sourcec.cfm.

Login Include

The login include file is used by unprotected pages to enable users to log in from any point (i.e., protected or unprotected) in the application. Without belaboring the point, the login include is very similar to the standard form-based authentication login. The essential code for this include, without additional markup and style elements, amounts to:

```
<form method="POST" action="loginRouter">
  <input type="hidden" name="j_security_check"
    value="/j_security_check"
  <input type="text" name="j_username">
  <input type="text" name="j_password">
</form>
```

Login Router Servlet

The Login Router Servlet is one of the pillars of active authentication, yet it's very simple to understand (see Listing 1).

The purpose of the Router Servlet is threefold. First, it sets session attributes for the values passed by the login include or by the default login page (login.jsp). Session values are set because other values, such as request attributes, do not persist across redirects. Second, the Router Servlet retrieves the originator of the request so that this page is reloaded when active authentication is complete. Note that this must be done differently for each of the possible request originators: the login include or the default login page. Finally, the Router redirects to the protected page Proxy.jsp, which will cause container-managed authentication to be invoked.

Login Proxy (Proxy.jsp)

The Login Proxy is likely the simplest component of active authentication. As mentioned earlier, the sole purpose of the Login Proxy is to keep track of which page originally requested authentication, and redirect back to this page when authentication and authorization are complete. The most important point to remember about the Login Proxy is that it must be identified as a protected resource using the Web application deployment descriptor. Once you've done this, the Login Proxy basically amounts to a couple of lines of essential code:

```
String redirectURL = session.getAttribute("originator").toString( );
response.sendRedirect(response.encodeRedirectURL
(redirectURL));
```

Default Login (Login.jsp)

The Default Login page is probably the most complicated moving part in active authentication, handling two very important responsibilities. First, this page needs to determine whether it was requested by another page within the application or if the URL was requested directly. Based upon these conditions, the Default Login will or will not render a selection mechanism that allows the user to choose where they should be directed to after authentication. Second, the Default Login may redirect to j_security_check for authentication, where applicable.

Before delving into the Default Login page, it's important to first understand how the page handles its different rendering states. Login.jsp is defined as both the form login

edit-on® JavaBean SDK. Look no further!

More than 1000 companies, universities and organizations worldwide have chosen the RealObjects edit-on® Pro XHTML/XML content authoring applet as their in-browser WYSIWYG editor of choice.

Based on the same proven edit-on® technology, RealObjects is now offering the edit-on® JavaBean SDK. It enables professional Java™ developers to enhance their AWT- and Swing-based applications with WYSIWYG rich text editing capabilities within minutes.

What you seek is what you get*

* RealObjects edit-on® JavaBean SDK enables WYSIWYG XHTML/XML content authoring. It brings word processor-like features to AWT- and Swing-based Java™ applications.

For more info and a free eval, visit:
www.realobjects.com

REALOBJECTS
Cross Platform Software

and form error page in the Web application deployment descriptor, using query string values of "action=protected" and "action=error", respectively, to denote conditions requiring initial or subsequent authentication. "Action=logout" will also be set by the application when a user wishes to log out. Given these three states, the Default Login page should render a selection mechanism to allow the user to select a page to be forwarded to upon authentication if:

1. The user is logged out.
2. The page URL was requested directly.
3. The page's action value is something other than "protected", "error", or "logout".

This determination is represented in Listing 2 by setting the isSearchSet variable to true if a page selection mechanism needs to be rendered.

If a page selection mechanism does need to be rendered, the j_username and j_password authentication inputs are rendered along with a mechanism that sets the value "ORIGINATOR" (see Proxy.jsp) to one of a series of page names for redirection after authentication. The action of this form should be the Login Router. A page selection mechanism should not be required for the form login or error page. In this case, a normal form-based authentication login group using the j_security_check action should be rendered.

The default login page's second yet equally important responsibility is redirecting to form-based authentication and removing the session's login attributes when the page is first loaded. If these login attributes are not removed, the application will enter an infinite redirect loop by continually posting invalid credentials. This can be avoided using the logic provided in Listing 3.

Despite the different approaches, the final result of active authentication is the same as the final result of form-based authentication. If the user is authenticated and authorized, he or she will be returned to the page requested. If not authenticated, the user will be returned to the Default Login page and an appropriate error message will be displayed.

Conclusion

When selecting an authentication mechanism for your next enterprise application, resist the urge to reinvent the wheel. With the exception of lazy authentication, form-based authentication is likely to be a suitable authentication mechanism for your application. With some patience and the active authentication components from this article, you can help form-based authentication overcome its lazy ways and meet your application's security requirements.

References

- Almaer, D. (2001). "Web FORM-Based Authentication." O'Reilly: www.onjava.com/pub/a/onjava/2001/08/06/webform.html
- Arumugam, P. (2002). "J2EE Form-Based Authentication." O'Reilly: www.onjava.com/pub/a/onjava/2002/06/12/form.html
- Bergsten, H. (2002). *JavaServer Pages, 2nd Edition*. O'Reilly.
- Hall, M. (2001). *More Servlets and JavaServer Pages*. Prentice Hall.
- *J2EE Platform Specification:* http://java.sun.com/j2ee/1.4/download.html#platform-spec
- *Java Servlet 2.3 Specification:* www.jcp.org/aboutJava/communityprocess/final/jsr053/

Forced to choose between HTML and Swing?

Are your users frustrated by clicking on page after page of your HTML web application?

Do you want your users to quickly navigate through thousands of records on the web?

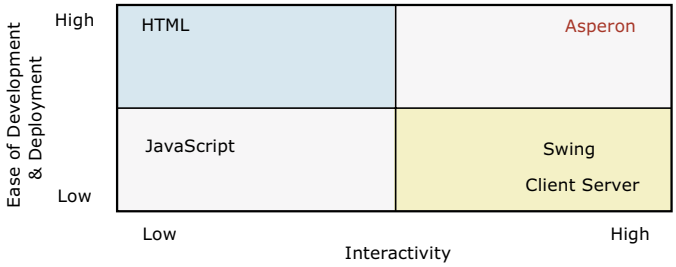
Are you facing deployment problems due to the need to install plug-ins when using Swing?

Do you want your applications to be accessible from any browser anywhere?

Now you don't have to compromise.

Get the best of both worlds with Asperon.

Asperon's next generation AppProjector™ technology includes a unique client that is as portable as HTML and as interactive as Swing. This allows you to quickly and easily develop web applications that have desktop-like user interfaces that do not require a plugin or client side install.*



- *Up to 10x faster than developing your own custom Swing client.*
- *Scroll through 10,000 records in seconds!*
- *Standards based - uses only Java/XML.*
- *Real-time dynamic updates from server.*

ASPERON™
www.asperon.com

* Works on most popular browers version 4.0 and above.

Download your free trial version today!

Jason Bell
J2SE Editor

JavaOne

The JavaOne conference passed me by this year, as did the previous seven. I never get the time to attend these things since I'm in the UK and it's a long journey. So I sat back in my big developer's chair and watched the Java world pass by like Weblogs in the night.

One of the hot blogging topics covered was, of all things, Christina Aguilera. More to the point, you could access vital Christina information via mobile technology. From a Java technology perspective, Sun hit the nail right on the mobile device head. From where I sat developers got the java.com site all wrong; here's a hint – *it's not for developers!* I was really happy to see the new site; Sun is finally marketing in the right direction for Java technology. They've figured out how to sell to the consumer. We can send the rocket scientists home now.

Figure 1 Christina checking out her own fashion tips

Java in the developer domain is all very well, but if it's not exposed properly to the consumer, then I may as well go back to Perl coding, as I won't have a steady job in 12 months.

The other hot blogging topic was Rave crashing on its first run. (I know this never happens to you guys.) Code will always work until it's shown in public; the general rule of thumb is the larger the public gathering, the larger the crash-and-burn element.

The blogging community once again has shown its major traits: first, as a community it can have the latest news up in a matter of minutes. Alan was blogging while the keynotes were happening; all I have to do now is discipline myself to read them. Now I don't want to lift our illustrious leader up too high but he does state facts and not opinions most of the time. This is a strong point. The second trait I noticed is a number of bloggers spend most of their time venting their spleen and stating opinion. I suppose there is an art to reading blog sites. I won't go on about the spelling mistakes (there are some blogs that read like neat Latin).

Were there any other highlights of JavaOne? The java.net community site was launched; I think it's a move in the right direction. I do wonder though how the well-established community will take to it. The first casualty I was notified about was that javagaming.org had moved its message boards to the new java.net site. There were many complaints that the message board postings were not available, but then all was well.

It was nice to see that James Gosling now has a blog of sporadic content. He also posted his blog software on the site (and the shell scripts to ftp/sftp the content; I can just see the Java purists staring into their coffee and muttering, "He used shell scripts," with complete distain). By doing this James also showed how to get the job done, neatly, cleanly, and the fastest route to a working model. So what if he uses shell scripts? It does the job perfectly and it inspires me to write some Windows batch scripts to do the same thing.

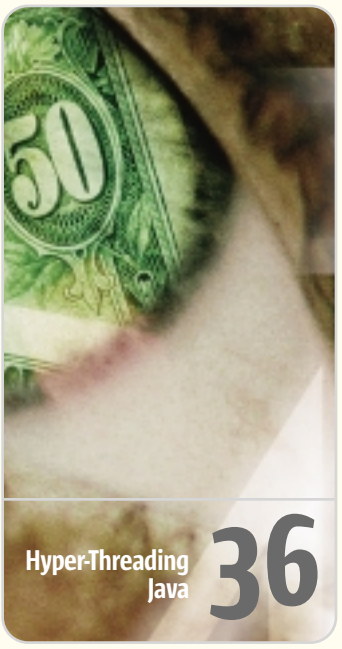
Blogs are here to stay but I have to question some of the content. As a reader I'm looking for fact, not opinion. At the time of writing there are only about five blogs I can say, hand on heart, that I read on a regular basis. Hopefully, reading the blogs of some of Sun's employees will reassure me that all is well in Javaland.

Blogs of Note

- *Matt Biddulph:* http://hackdiary.com
- *Edd Dumbill:* http://usefulinc.com/edd/blog
- *Russell Beattie:* www.russellbeattie.com/notebook
- *Charles Miller:* http://fishbowl.pastiche.org
- *Mark Pilgrim:* http://diveintomark.org

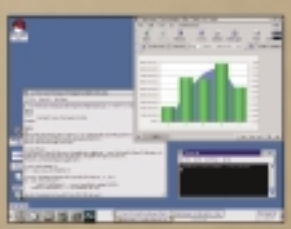
Jason Bell is the senior programmer for a B2B portal. He's also a keen supporter of people reading the API docs before asking questions. In his spare time he's involved with building RSS development tools.

jasonbell@sys-con.com



Throwing Our Hat In The Ring...

...with a whole new brew of Chart FX.



10 Years of creating quality developer charting solutions gives us an edge over others in the industry. So, although we're the new kid on this block, we've been in the neighborhood for a long time. The maturity of Chart FX and our support will prove it. **Chart FX for Java** is a 100% Java component that uses your J2EE application server and JSP technology to produce charts in a variety of formats: PNG, JPEG, SVG and Flash. Additionally, it will produce native Windows output in the form of ActiveX or .NET. Server-side Bean or Enterprise Java Bean (EJB). Data population through JDBC, XML, text files and more. ***For more information or to download a trial version, visit www.softwarefx.com.***

Chart FX
for JAVA

Learn more at
www.softwarefx.com

Win a 2003 Honda Shadow Sabre!

< visit www.softwarefx.com to enter - no purchase necessary >

Java Games Development

Part 1 of 3 – Let the games begin

hosted by
Jason R. Briggs
jasonbriggs@sys-con.com

I recently had a hankering to play an older (not ancient) PC game that I used to enjoy. Since I've moved my entire desktop over to Linux (for almost a year ago now) that meant stealing my wife's Windows laptop and trying to install the game on that. Two FDISKs and one Windows reinstall later the laptop's HDD is still woefully short of a game, and I'm back at square one.

This isn't an isolated case – there's a whole backlist of games that in an occasional retro mood I've felt like having another go at…even to the point of attempting to get some of them working in various DOS/Windows emulators on my Linux machine (with only limited success), because it's just as much of a pain in the proverbial posterior to get them running on later versions of Windows as on what the games were originally intended for.

I would even be tempted to dish out a few bucks to buy retro games, if I could only be sure that they were actually going to run successfully.

How much easier would this process have been if some of these games had been written in Java? The sad fact is a lot of them could have been developed in our favorite language had Java 1.4 been available 3–5 years ago. Assuming there wasn't a huge reliance on quirky native bindings, they would probably be relatively easy to get running now. Alternative platforms needn't be a huge issue either (also assuming that those native bindings were available on your platform of choice).

All of this had us thinking at *JDJ* (yet again) about games development in Java. Some might say it's flogging a dead horse; some would just wonder why, despite the cross-platform and preservation of investment advantages I've just mentioned. Which is the reason we've gathered together an eclectic group of interested parties to discuss that very topic – Java games development.

The players are:

Jason R Briggs: *Java Developer's Journal* contributing editor and your host; games player when he has time, games developer…on occasion.

Gerardo Dada: Metrowerks' product manager for CodeWarrior Wireless Studio.

Erik Duijs: Former musician/engineer/producer with a (games) programming passion, now an IT consultant. Switched careers for the sake of better pay as well as maintaining a passion for music instead of "eating it" so to speak. Author of the Java Emulation Framework (JEF) and CottAGE.

Shawn Kendall: Developed Java and Java 3D–based game technology demos for Full Sail, Sun Microsystems, and I.M.I., and displayed at GDC and SIGGRAPH since 1999. He has five years of 3D technology teaching experience, and in 2002 founded Immediate Mode Interactive, LLC, a game technology company dedicated to the development and application of Java technology in computer games (www.imilabs.com).

Jeff Kesselman: Architect for game technologies, Advanced Software Technologies Group at Sun. He worked on the JDK performance tuning team and co-wrote *Java Platform Performance: Strategies and Tactics*.

Chris Melissinos: Sun's chief gaming officer and responsible for driving an industry-wide movement toward Java technology–based game development and building infrastructure programs for massively connected game play.

Caspian Rychlik-Prince: An IT consultant in the UK who for the last 10 years has specialized in client/server systems with RDBMS back ends. Since the contract market in the UK has all but dried up, he has shifted focus to creating assets rather than doing a job, so he's writing a game now.

Doug Twilleager: Chief architect of the Java Games Initiative at Sun Microsystems. One of the architects of Java 3D, he has also worked in the graphics research group at Sun looking at advanced rendering techniques and programmable shading.

David Yazel: VP of software development of trading systems and portfolio management systems at a leading financial investment company (by day), and by night a games developer for (and founder of) the Magicosm project (a 100% Java-based MMORPG).

JDJ: C and C++ have the market (and mind) share in terms of games development languages. Indeed, the mainstream games market has traditionally been very reluctant to change development paradigms. What do you think needs to happen to shift that mind share toward other languages/platforms, such as Java?

Jeff K: It is important to remember that we were asking almost the exact same question about C++ less than 10 years ago. And before that it was C versus assembly code. More recently, scene-graph systems such as RenderWare were fighting the same perception/adoption battle.

In the end, nothing succeeds like success. A hit game ships in pure Java and it will put all the nay-saying to rest. To get there though there are many things that can help. Outreach and education are two of them. Our industry runs to a large degree on "common knowledge." Insofar as it's wrong (e.g., "Java is slow"), we can educate and help change perceptions. Part of this is explaining why these preconceptions are wrong; the other part is proving the point with working demos. We can also explain the perceived and less-known advantages. Portability is the one everyone usually thinks of, but there are perhaps even stronger advantages in the areas of productivity and code-correctness.

Finally, we need to unsell some of the oversold aspects of Java – portability being a good one, in my opinion. One-hundred percent portability is not only unlikely but in most cases undesirable for games that should ultimately be tuned for each platform. I personally think 85% portability is a more sensible and reachable goal.

Doug T: One of the biggest obstacles that desktop Java faces right now is the ability to access the latest game technologies in a standard way. There are lots of community supported solutions for things like 3D, audio, and input device

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**

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	20GB	55GB
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 + ASP.NET	x + ASP.NET
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.

support. However, until there is a widespread standard available across multiple platforms, developing desktop clients will be painful. These standards don't need to come from the Java Community Process; de facto standards work just as well. There are, of course, other game areas where Java is gaining acceptance. An example of this is mobile device games.

Gerardo D: From my perspective there are three things that must happen to change mind shares toward Java. One is performance – game developers are all about optimization: getting one more frame per second allows them to add more details and textures, more calculations to the physics involved, and more realism and improve the overall game experience. Java is still perceived as slower than compiled C/C++. Perceived is the key word here – there are AOT compilers for Java and DAC technologies that reduce the performance hit of having a virtual machine. If you ask game developers who develop games for wireless devices, a J2ME game usually performs faster than a C++ version of the same game running on BREW.

Number two is access to low-level stuff. With C++ you don't have to wait two years for a JSR and an implementation to use a low-level function. Console manufacturers try to differentiate and add new technologies for

developers (like vectorization) that would be hard to use from Java. Developers want to use maximum performance and all the features of a console or a PC. An option would be JNI.

Three is industry support – outreach and education like Jeff mentions, but also things such as middleware technology. As games become more complex and platforms have higher performance, developers rely on third-party technologies such as Physics engines, AI, and human rendering.

I also agree with Jeff; once you have a success story – a developer who builds a great game and leverages the portability of Java to reduce the investment and time-to-market to offer it in multiple platforms (PC, Mac, PS2, etc.) – then the industry will start paying attention.

Jeff K: Actually that is not strictly true of modern desktop VMs. About the only thing AOT compiling gives you in the desktop space is faster start-up. AOT compilers by all tests I've seen are slower then JITs in an actual run, or, if they manage to reach JIT speed, do so at the expense of a seriously bloated footprint. To put it simply, this is because a JIT adjusts the code generated at runtime according to the actual runtime environment (e.g., processor type) and how it's actually used (e.g., in-lining methods that are "used" monomorphically whether they are provably defined that way or not).

I am specifically excluding J2ME here, because obviously in the tiny memory and highly constrained environment of current cellphones you can't effectively deploy a JIT solution, and here (and only here) I would agree that an AOT compiler might make sense. However, this is a temporary phenomenon as we are seeing cellphones quickly rise to PDA level and PDAs quickly reaching the size of set-top or small desktop systems. It's reasonable to assume this trend will continue for awhile at least.

> "We need to unsell some of the oversold aspects of Java – **portability being a good one, in my opinion**"
>
> –Jeff Kesselman

Cas P: I have to be very careful answering these types of questions. Like a lot of engineers I have a binary world view, but I'll try to be reasonable.

The C/C++ market share is a product of many factors. The first and foremost is inertia: all the people currently in the industry who are now experts at what they do write in C++. The C crowd is actually dying away because the advantages of C++ are difficult to ignore. But we all know that once we get comfortable with something and good at using it we have a very strong tendency to stick with what we know – and that's actually very wise. I suspect we will have to wait a long time for the veterans to gradually leave the industry and take their own ways with them.

The second factor is higher education courses. A lot of people who graduated from university in the past 10 years started out on something useless like Modula-2 to learn programming and got into C++ as their courses progressed. They left university with a working knowledge of C++ and were very likely exposed to MSVC6. That gives you a huge pool of C++ programmers. Nowadays, a lot (most?) of the C++ courses are being replaced with Java – and if not Java I understand Delphi is very popular (also for good reasons). In the middle of the next decade a fresh-faced C++ programmer will be a rarity, mark my words. However, it does mean that the vast majority of Java programmers out there are really inexperienced and won't have very many experienced Java programmers to learn from. This is the ascendency. As Jeff alludes, it took 10 years for C++ to finally oust C. However, it would probably be a true case of hype if we were to claim that Java's time is right now – it's not; it needs that 10 years. I think there may be a correlation between 10 years and the length of time the average programmer remains in the games industry. It's a generational thing. Right now I think we're three years in: by 2010 the majority of games will probably be written in Java, but only if certain things are done to Java to make it as flexible (and indeed, as powerful) as C++.

Apart from waiting, the real disadvantages of Java would have to be addressed before you see any significant shift in mind share occurring. I'll stick my neck out here and say that the frequently quoted productivity benefits

Application integration is an intimidating challenge for any enterprise; the downtime costs of unreliable integration are unthinkably expensive. To keep profits climbing, enterprises are adopting open standards-based messaging, caching and integration solutions.

The SpiritSoft framework integrates your legacy applications into a single JMS-compliant environment. SpiritSoft technology goes beyond JMS to give you a secure, scalable and robust enterprise-integration strategy to provide for your future messaging and Web Services requirements.

Secure your integration foothold … there's no room for error.
Download our white paper www.spiritsoft.com/climber

spiritsoft.
go beyond jms

of programming in Java are largely nullified when writing games because of the horrendous amount of work involved in getting performance and quality out of the finished product, right down to writing OpenGL wrappers up through fooling around trying to fill structures in buffers without direct language support, through avoiding the wrath of the garbage collector. That's just one of the technical disadvantages. It also happens to be a performance disadvantage in a particularly critical section of a game loop, and that's going to be picked up on very early by anyone trying to do something cutting edge.

Jeff K: I would classify this as "maturation" and a learning curve myself. In the maturation category is the OpenGL wrappers you mention. It's insane that everyone currently feels a need to do this themselves. I'm convinced standards would emerge under any condition, and the conditions are actually looking good for this to happen fairly soon.

In terms of taming and controlling the GC, that's a matter of the learning curve. As in any language, you develop habits in how you use it as you come up to speed.

I actually think it's remarkable if Java's advantages in performance even end up making the learning curve a

net-zero cost. From what I know, it would be the first time that was true for any new language or environment and Java is both at once.

Cas P: Another disadvantage is dependency on the correct VM being installed on the target platform. If you were to find Java on the next PlayStation (so help me, but if it has OpenGL on it too I'll be over the moon), you'll be fine, of course, because it'll be guaranteed available. But for smaller games, the VM overhead is too great to be a reasonable proposition. That doesn't necessarily mean you can't use Java, but because of a curious licensing restriction it does rather restrict what bits of Java you use.

This brings me to a remedy for that disadvantage – the "J2GamingEdition" – which is almost but not quite what we have with the J2ME CDC profile. What I'd really like to see in terms of licensing is that the Java 2 platform is cleanly separated from the Java 2 runtime environment, with only java.lang being necessary to run the JVM. This gives the developer the feeling of being close to the metal, mean and lean. It's a crucial comfort factor in programming games to know that you can solve any problems by yourself through programming, not waiting. It's bad enough waiting for OpenGL drivers that work and hoping there are no VM bugs without dicky implementations in the java.* libraries causing a performance hit or bugs. It's particularly important to stress how little of the standard Java libraries are actually useful in games programming. One of the reasons Java is so productive in the normal application space is that it has a rich set of easy-to-use features. But use them in a game and you discover why they're easy to use: they're not tuned; they're very inefficient in most unexpected ways. No doubt more on this topic later.

Jeff K: Licenses aren't set in stone. With good enough reasons, corporations can change them (in point of fact, in one-off ways Sun already has). Still, I agree 100% that one of the bigger barriers right now is the unfortunately restrictive Java license terms and I'd personally like to see that change.

In terms of the standard Java libraries, here I would disagree. It's the old 90/10 rule. Ninety percent of your work is done by 10% of your code. You can afford to trade efficiency for speed of coding and correctness of results in many places. (This is formalized in many games by creating a "scripting" layer that in fact is dog-slow in raw speed terms.) Even where you need to tune, premature optimization often creates more trouble than it solves.

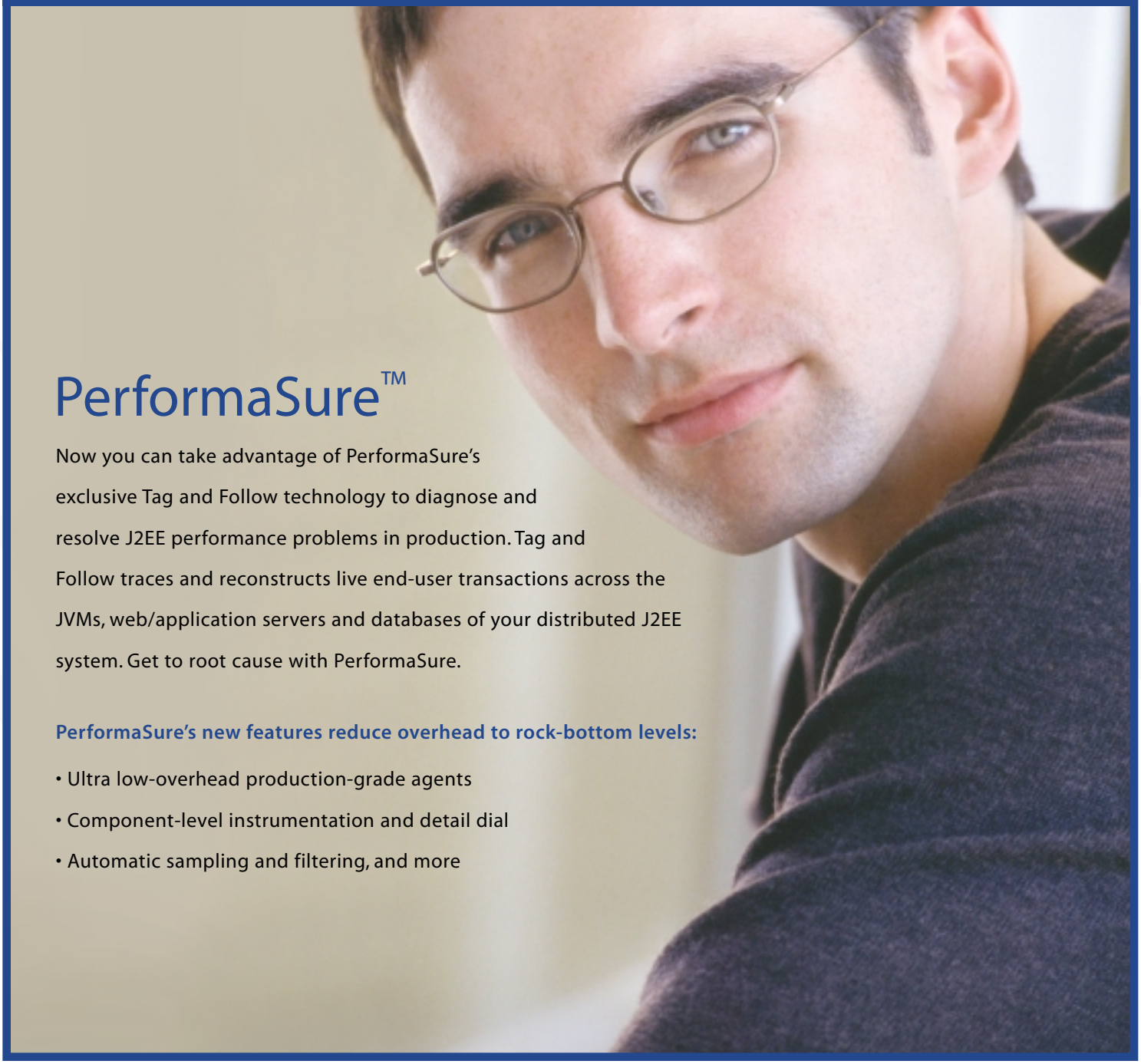
A classic example that gets worked over and over again is the Java collection classes and the ways they create garbage. When I use collections I never specify specific types (e.g., ArrayList). All my variables are of Interface types (e.g., list). This allows me to easily tune on two levels. If I find a linked list is really more appropriate by the time I'm done, I can change it by simply changing the type after the new one. But if I find none of the supplied implementations work the way I need, I can in fact write my *own* special case implementation at that time.

I would argue that this all still results in major time savings over guessing at the beginning of my coding what the ideal data structures would be, and then either living with a bad guess or doing major code rework. Finally, I personally think too *much* emphasis is placed on the high-level Java features when talking about coding efficiency. There are more fundamental ways in which Java promotes proper code and accelerates its development. Most of these can be classified under one key Java design goal: wrong code should fail early.

> "By last year, every Java IDE out there had surpassed the incumbent **C++ IDEs in terms of functionality and ease-of-use**"
>
> **–Caspian Rychlik-Prince**

There are whole classes of C/C++ errors that just won't compile in Java (e.g., using uninitialized variables). There are many other common ones that will throw an exception right away in Java, whereas in C they hide till late in development or worse get out into the field (e.g., null pointer access, array overrun/underrun, etc.).While I agree that a lot of the Java libs aren't useful for games, I think the problem is that they aren't designed for what we do and thus provide functionality that isn't useful. (Anyone writing a game client

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

QUEST SOFTWARE®

Take a tour of PerformaSure today:

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

that uses SQL?) But I don't think it's as big a hit on the Java productivity story as some folks might first assume.

Cas P: For sure, and this is its best feature. Currently though I'm finding some classes of error difficult to deal with, like bunging data in the wrong place in a DirectByteBuffer because of a language hindrance. It's swings and roundabouts, but more swings in general, and that's a good thing.

A side effect of this is that the IDEs in Java are of absolutely stunning quality. By last year, every Java IDE out there had surpassed the incumbent C++ IDEs in terms of functionality and ease-of-use. Eclipse is taking on MSVC head on and winning by a long shot. This is a great thing; for years the de-facto hard-core Windows development tool was MSVC, and the other C compilers were for eccentrics and specialists. We have a situation here in which there are at least five different major IDEs, and because they all produce identical code, there is absolutely no reason why a team couldn't use all of them to write a game. I've always found it rather like giving a bunch of artists a set of paintings to create and then saying they all have to use the same sized brush and the same three paints.

But the end result is the same – it's a big, big chink in the Windows monopoly's armor. M$ doesn't really have a chance to compete any more, and this is quite remarkable.

Back on topic for a moment, start-up time is another issue for Java. It's actually quite acceptable in the client VM, but performance is very disappointing. Start-up time in the server VM is terrible, but performance is the equal of C++ in many areas (it's currently still much worse, 50% at best, at floating point and memory access though – and that's unfortunately really where it counts in games right now). Again, "J2GE" could do with a hybrid VM reference implementation that perhaps did a two-phase compilation. These ideas have been talked about for at least two years now but we come to another stumbling block: Sun doesn't have the resources and certainly doesn't have the will to make it happen.

Jeff K: Maybe I'm an old fogey but start-up time still seems pretty fast to this kid who grew up with a C64. I do agree, however, that it's unfortunate that *nothing* can be done during the load-up. If it were my commercial game I'd probably write a machine-specific bumper that has a splash-screen and some animation up on screen, then starts the VM.

Also, as an aside, Apple's been doing interesting things with start-up time in their version of Hotspot.

Erik D: I think execution speed is a very overrated argument for using C++. As game projects grow more complex, the language should keep up in order to minimize production costs. I notice that it's getting increasingly likely that when you buy a game, it's going to be more buggy than the previous game you bought. This is directly related to the problems C++ has in terms of production costs; problems that Java can help address. The few frames per sec you might lose with Java is not really important. I think Java's speed is acceptable and getting better. GC issues are there, but Unreal also has GC, which is sometimes clearly visible and nobody complains. Of course, this doesn't mean it shouldn't be addressed, but I don't feel it's a valid argument against using Java.

Most of all, right now we need some kind of Carmack that proves this point to a broad audience in a Wolfenstein3D/Doom kind of way. I feel that's much more important than solving technical issues, which I believe we can already work around.

Shawn K: Shifting mind share is simple. It is *not* a technology question. It is a "proof (for developers) and money (for publishers/managers)" question. All that's needed is one big, high profile hit. That's all. Many developers scoffed at RenderWare in the beginning, but when huge hits like Tony Hawk 3 and GTA3 (Grand Theft Auto III) were released, they really put it on the map.

Now, how to make a hit with Java? That's an entirely different question. As a quick high-level answer I'll say it will take all the things that are needed to make a game a hit regardless of language, as well as significant (read $$$) changes to Java for consoles, in addition to some serious support for J2SE.

I was at the Digital Media Alliance of Florida meeting recently, here in Orlando. The panel discussion had a lot to say about independent developers, a place where game developers are willing to take risks on game design and game tech… Even though one of the represented companies there was an independent, the general consensus was that indie development is going away unless some kind of big indie festival with contract awards is created, because publishers will not/cannot afford to back experimental games *and* experimental tech. It's that simple. The industry is more hit driven than ever due to the fact that more people than ever have multiple consoles, and the majority of players are only buying AAA titles now.

The buying characteristics have changed from the one console days. Players would buy an average of 5–6 games a year, and 1–2 of those would be second-tier indie-type games. Now, you can buy all 5–6 as AAA titles because you get big games but on different platforms. Publishers know this and are shying away even more from small production, indie-type projects. Add to that the much higher cost of developing current 3D games as compared to the 2D world of the past, and the risky indie market gets squeezed to a speck.

This discussion on Java games will continue

SEIZE THE POWER

ORACLE WORLD

Why James keeps coming back: ***"OracleWorld is great for networking and establishing relationships with Oracle experts and users, along with learning about the latest industry trends and developments."***

James Holmes
Java Development Consultant

ATTENDEE SINCE
2002

This year at OracleWorld find out how Oracle powers the Grid — and what this means to you. Join us as Larry Ellison unveils Oracle10G — the first database designed for emerging grid computing.

Get the information you need at OracleWorld.

REGISTER TODAY!
oracle.com/start
keyword: OW2003
1.800.355.5808

September 7-11

Moscone Center, San Francisco

ORACLE®

Platinum Sponsors

DELL™ hp invent intel® Sun microsystems We make the net work.

Premier Sponsor

EMC² where information lives

Hyper-Threading Java

SMT technology is here to stay

by Paul Bemowski

n early 2002 Intel became the first chip manufacturer to release a processor incorporating a new technology known as Simultaneous Multithreading, or SMT. Intel's SMT implementation (dubbed Hyper-Threading or HT) has been available in their Xeon processor line for over a year, with little fanfare. In April 2003, Intel announced that HT technology will be added to its desktop-focused Pentium 4 line of processors. With HT enabled on one of these new systems, the BIOS will present a single processor to the operating system as two logical processors.

As Java developers, we should all be excited about this new feature of Intel processors. The java.lang.Thread object was one of the key factors driving Java to the strong position it enjoys in the server-side applications market. Both client and server applications written in Java often make heavy use of threads. Indeed even if an application does not use threads explicitly, all JVMs will use at least one background thread – the garbage collector. SMT holds the promise of significantly increasing Java's server-side performance by more completely utilizing existing processor cycles in multithreaded applications.

This article attempts to explain the concepts of Simultaneous Multithreading in layman's terms, presents the development of an *n*-thread benchmarking suite, and uses that suite to produce concrete results of multithreaded benchmarks on HT and non-HT systems. We'll investigate various operation types to determine the factors that affect Java performance enhancements on Hyper-Threaded processors. Finally a series of conclusions and speculations are derived from the data collected.

Understanding Symmetric Multithreading on Intel Processors

Intel processors with HT technology carry two copies of the processor's architectural state on the

same chip. This second architectural state stores a second thread context. Conceptually, this type of processor architecture splits each physical processor into two or more logical processors. Physical SMT processors present themselves to the operating system as separate logical processors. As we'll see later, it can then become important for the operating system to be aware of and to differentiate between logical and physical processors. Figure 1 illustrates the difference between SMT and non-SMT processors.

What is the benefit of SMT? As it turns out, the more expensive processor resources can find themselves underutilized while an active thread performs long latency operations. A cache miss, for instance, will require the processor to make a request to main memory. The majority of the processor's resources remain idle for this period of time; however, the processor presents itself to the operating system as busy. SMT systems use this slice of time to execute the operations of another on-chip thread context.

SMT processors contain an onboard scheduler to interleave multiple threads operating on the physical processor. If a thread encounters a long latency, the processor will immediately execute the instructions of the second on-chip processor state. For two threads accessing the same processor resources, the onboard scheduler will interleave the threads much the same as a software thread scheduler. This interleaving has a small amount of overhead, which can decrease the efficiency of the processor in certain situations. On an aggregate basis, however, processor performance is increased.

Using SMT it becomes apparent that depending on the work that each thread is doing on adjacent logical processors, we could see performance increases or decreases. Various papers (see references) studying multithreaded performance indicate generally positive results, with some research indicating perceived performance gains as high as 50%.

HT-Enabled Systems

Intel Hyper-Threading requires support from three fundamental components of a system:

"The more expensive processor resources can find themselves underutilized **while an active thread performs long latency operations**"

1. The processor
2. The chipset
3. The operating system

Processors Supporting HT

Hyper-Threading was incorporated into the Xeon class processors in early 2002. Xeon is not to be confused with Pentium III Xeon. When Intel changed the Xeon's core to P4, it dropped the P4 designation, calling the processor simply Xeon. Recently, HT has found its way to the desktop P4 processor. Not all processors in each of these processor classes are capable of Hyper-Threading, however.

Table 1 indicates which processors support Hyper-Threading. The table also indicates factors that you can use to determine whether a given Intel processor supports HT.

With the release of the 3.06GHz Pentium 4, Intel changed the P4 logo, incorporating the letters H and T to indicate that it's a Hyper-Threading processor.

All recent Xeon processors support Hyper-Threading, but again, be sure to watch out for the 256KB L2 Cache version, which does not.

Chipset Support for HT

Not all chipsets support HT. Check with your chipset manufacturer to ensure that you can enable and disable HT support via the BIOS.

All HT chipsets interleave processor numbering to help less sophisticated thread schedulers make complete use of available physical processors. The chipset will present the logical processors to the OS as follows:

```
Logical CPU0 = Physical CPU0, Logical CPU0
Logical CPU1 = Physical CPU1, Logical CPU0
Logical CPU2 = Physical CPU0, Logical CPU1
Logical CPU3 = Physical CPU1, Logical CPU1
```

Operating Systems Supporting HT

Given a processor and chipset that support Hyper-Threading, the operating system must also be HT aware. Table 2 shows the OS support for several currently available operating systems commonly run on Intel-based hardware.

Windows

The Windows 2000 operating systems do not differentiate between logical and physical processors. Therefore a 32-processor HT system will support only 32 logical processors. It will work; however, the additional processor resources will not be utilized.

Windows users should check software licensing agreements to confirm that they recognize logical processors. Generally XP will support licensing on a per physical CPU basis, while Windows 2000 will see logical processors as physical processors for licensing purposes.

Figure 2 shows a Windows XP Pro task manager on a dual-processor HT system, note the four distinct "CPU Usage History" charts depicting the four logical processors.

Linux

The 2.4 kernel began supporting Hyper-Threading on the Intel Xeon processor as of version 2.4.18. The thread scheduler in 2.4, however, does not understand the difference between logical and physical processors, in addition to many other SMT scheduler optimizations, similar to the Windows 2000 family of products. This can lead to degraded performance in situations where two threads are scheduled concurrently on one physical processor, while the other physical processor is left idle.

As of kernel version 2.5.32, the thread scheduler was updated with advanced features to support Hyper-Threading. The 2.5.x kernel is the development branch that will become the 2.6 kernel. The exact release schedule for 2.6 is unknown, but in a recent interview Linus Torvalds indicated that 2.6 would likely be released in Q4 2003.

Figure 3 shows a Red Hat 7.3 installation running the 2.4.18 kernel with Hyper-Threading enabled on the system. Note the four CPU states indicated as CPU0-CPU3 on top. Also note that CPU0 is running at 100.1% utilization – wow, Hyper-Threading is cool!

Processor Class	Speeds	Determining Factors
Pentium 4	3.06GHz	Only the 0.13 micron, 3.06GHz, P4 with a 533 or 800 MHz system bus
Xeon	1.8 to 3.06GHz	Only Xeon's with 512KB L2 cache support HT
XeonDP	2.0 to 3.06GHz	Only Xeon DPs with 512KB L2 cache support HT.
Xeon MP	1.4 to 2.0GHz	All support Hyper-Threading

Table 1 Intel processors that support Hyper-Threading (as of this writing)

OS	HT Support Physical	CPUs
Windows XP Server	Supports HT	32
Windows XP Pro	Supports HT	2
Windows XP Home	Supports HT	1
Windows 2000 Server	No differentiation between physical and logical processors	32
Windows 2000 Pro	No differentiation between physical and logical processors	2
Linux, 2.4.x kernel	Supported from version 2.4.18. No differentiation between physical and logical processors.	32
Linux, 2.5.x kernel2	Enhanced thread scheduler available from 2.5.32.	32

Table 2 Operating system support for SMT

Operating System	Windows XP Pro	Red Hat Linux 7.3 with 2.4.18
Processors	2x Pentium Xeon, 2.4 GHz, 512MB cache	2 x Pentium Xeon 2.4 GHz, 512MB cache
Memory	1GB	2GB
Disk	60GB, 7,200 RPM, Ultra ATA 100	2 x 72GB, 10,000 RPM SCSI

Table 3 Comparison of Windows XP Pro and Red Hat Linux 7.3

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

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

Not any more.

Get Wily.™

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

wily technology

Threaded Benchmarking on HT and Non-HT Systems

Our goal here is to understand the effects of Hyper-Threading processors on the performance of multithreaded Java applications. To do this, we need a test bed that will allow us to execute heavily threaded operations and track performance variations against thread count in HT and non-HT systems.

Thread Bench Design

At a basic level, the test bed should be able to execute multiple operations across *n* threads, observing the total throughput of operations per unit of time for a run. On a dual-processor system, we should see nearly double the performance on a CPU-intensive operation using two threads instead of one. The performance of CPU-intensive threaded operations on HT systems will vary based on the operations and the level of concurrency possible on a single physical processor.

Our focus here is to explore which types of operations will and will not benefit from HT technology. Given this we need to be able to quickly implement and test multiple types of operations.

There are several Java benchmarking systems available on the market. Many are older and focused on applet performance. Some newer benchmark systems like VolanoMark or SPECjbb2000 test the threaded performance of systems; however, they don't allow us to customize and focus on specific individual operations that could affect performance on an HT system.

These requirements drove the design and coding of an *n*-thread Java benchmark framework. The framework supports pluggable operation classes and produces plottable results for a range of thread counts from a single test suite execution.

Figure 4 presents a functional/UML diagram for the system design.

The resulting benchmarking framework has the following features:

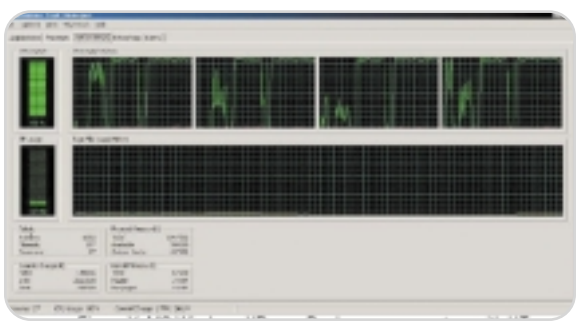


Figure 2 MS Windows XP on a dual-processor system with HT

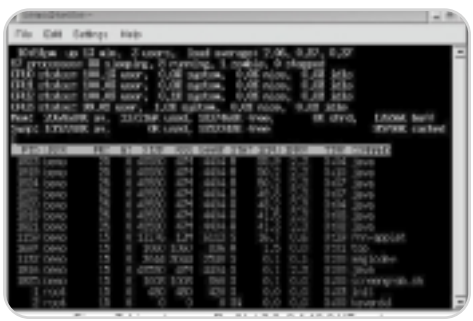


Figure 3 Linux on a Red Hat 7.2, 2.4.18-3 HT system

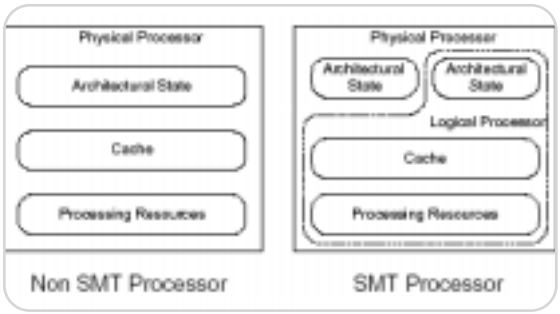


Figure 1 SMT vs non-SMT processor architecture

- ***Initialization of operations on the JIT:*** Modern JIT compilers will optimize "hot spots" in the code. The performance of any given operation will improve over the life of the VM, so the ThreadBench framework gives operations a chance to initialize on the JIT before the tests commence.
- ***Operation abstraction:*** By developing a generic operation interface and using dynamic class loading and initialization of the operation to be tested, we can quickly prototype and test various processor-intensive operations.
- ***Test suites:*** Using test suites, ThreadBench runs a given operation configuration through several iterations of the test with different numbers of threads. This allows a series of tests to be repeatedly run on several machine configurations with minimal effort.
- ***Multiple runs:*** To smooth out anomalies in the test, each data point is created by averaging data from several runs. This is configurable; some tests have a larger standard deviation than others.

The code for this article can be downloaded from the ***JDJ*** Web site, www.sys-con.com/java/sourcec.cfm.

Factors Affecting Performance

Use of Threads

This seems obvious; however, it needs to be mentioned: single-threaded applications (often client applications) will see little performance gain. Server-side Java applications make extensive use of threads, making them excellent candidates for performance improvement from SMT.

Nonthreaded applications may still see some benefit. Java's garbage collection and background JIT compilers operate as daemon threads in the local JVM. In addition, concurrent processes could make use of the additional processor resources.

The Operating System's Thread Scheduler

In an HT system, a single physical processor is presented to the OS as two logical processors. This requires the OS to differentiate between physical and logical processors and make intelligent decisions about thread scheduling.

The thread scheduler on a dual-processor HT system will see four logical processors. A poor thread scheduler could schedule two CPU-intensive threads onto separate logical processors representing the same physical processor. This would result in a perceived performance decrease on an HT-based system.

CPU Resource Utilization

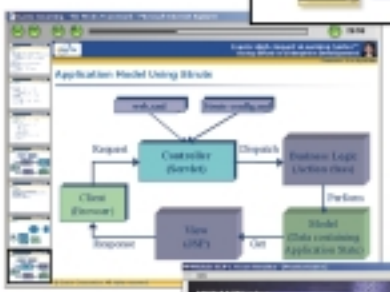
Hyper-Threaded processors do not duplicate all available resources. Two threads performing fundamentally similar operations on separate logical processors will likely see little performance gain. For HT to be a benefit, the two threads coexisting on a physical CPU must perform a variety of operations to allow the processor to make better use of latency.

isavix.net

Innovative "learning and sharing" solutions for **Java** developers

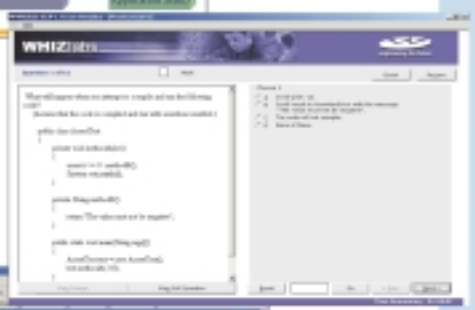
Searchable Books

- Conduct searches across 1500+ technical books from publishers such as Addison Wesley, Microsoft Press, O'Reilly, Prentice Hall, and many more. Zero in on answers to time-critical questions instantly.
- Read the books on your Bookshelf from cover to cover. Or, simply flip to the page you need.
- Browse books by category. Researching any topic is a snap. From XML, to database to .Net, you'll find your answer here.



eLearning Courses

The Isavix High-Impact eLearning Series™ is designed to deliver online training in the most efficient manner possible. Each course is approximately one hour in length with no "homework" assignments and can be viewed in Flash and PDF formats. Sample courses include The Struts Framework, Introduction to Apache ANT, ASP.NET Crash Course, and more.



Certification Exam Preparation

Whizlabs exam simulators are available for Sun, IBM, BEA, CISCO, Microsoft, Oracle, CompTIA, CIW and Red Hat certifications. These products have been developed by experts who are considered the authorities in their subjects.

From computer professionals to Certification Training centers to College and University students, to hundreds of corporations ... the Whizlabs solutions have helped over 150,000 people get their certifications.

FREE Resources and Tools

isavix.net is a site that provides eLearning, information sharing, software exchange, helpful tools, and valuable links for emerging technology (e.g. J2EE, .NET) professionals.

Our free tools and resources include Discussion Forums, Code Exchange, My Secure Files, Quick Polls, My Bookmarks, Download Links, My UML Diagrams, My eLearning, and much more.

Custom Community Portal

For Your Organization

If you like the many benefits isavix.net provides but would want a custom version of our portal available exclusively for your organization, we can do that for you!

Our community can either be installed within your organization's Intranet or provided to you as an ASP solution.

The benefits include an enhanced user experience, professional development, improved team dynamics, effective project management, and better resource allocation. The ROI for an organization is high because developers can save from a few minutes to hours a day since the relevant information is readily available to them, when they need it!

Corporate and Government solutions available ... contact us for details.

isavix
Delivering Emerging IT Solutions

Isavix Corporation
Herndon, VA
1-866-472-8849
info@isavix.com

"Single-threaded applications (often client applications) will see little performance gain"

Performance of Threaded Benchmarks on HT and Non-HT Systems

Tests were run on two HT-capable dual-processor systems (see Table 3).

Hyper-Threading requires BIOS support, making it easy to enable and disable the feature in the boot setup program for various runs.

Each test was run with the Sun JDK 1.4.1_02, using the –server flag on the Linux and XP systems. Tests were also run with the IBM 1.4.0 JVM, with no command-line flags, on the Linux system.

The tests devised are by no means comprehensive. The goal was to stress the processor, using different processor resources, to try to gain some insight into the effects of SMT processing. The series of tests was run on each of the above systems, with and without HT enabled. Each of the operation algorithms tested is briefly described, followed by results and some discussion and interpretation.

Note: To save space, the XP and Linux tests are shown on the same plots. The data *should not* be directly compared, however. The tests were run on different physical hardware, indeed the processor speeds on the XP machine were higher than on the Linux machine.

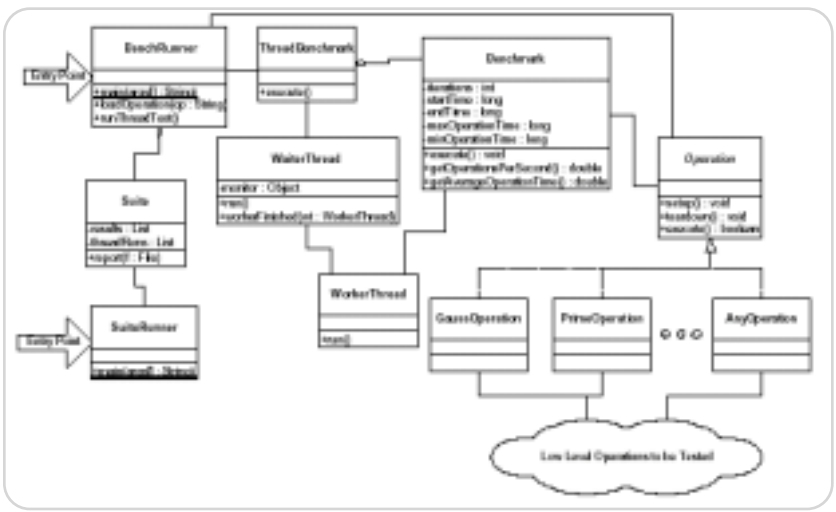


Figure 4 Pseudo UML for the threaded benchmark framework

Test 1: Gaussian Elimination, 500x500 matrix (Floating point intensive)

Gaussian elimination is a very common algorithm used to solve systems of linear equations – a common task in finite element applications, weather simulation, coordinate transformations, and economic modeling among other things. Algorithmic optimizations are often done for sparse/banded matrices; however, the core of the work is fundamentally the same – large numbers of floating point calculations are required.

To simulate this, a Gaussian elimination algorithm with scaled partial pivoting and back substitution is used (see Figure 5). A full matrix is constructed of random doubles using Math.random(). The population of the matrix is carried out in the setup() method and is not considered part of the operation.

This operation carries out large numbers of simple floating point operations on doubles. All calculations are done in the Java call stack, though it's highly likely that the code was optimized by the JIT before the tests were run.

It seems that this operation does not scale well into threads on any JVM. The Sun VM on Microsoft with Hyper-Threading does significantly worse than the Linux JVMs with or without Hyper-Threading. There are no synchronizations in the operation whatsoever. Poor scaling into threads could be due to memory barriers, or contention for a bus or main memory.

Test 2: Calculation of 2000! (Integer intensive)

Calculation of factorial (! operator) is used often in probability calculations. It's used as a portion of the formula for combinations and permutations. Factorial is defined as follows:

$$N! = 1 \times 2 \times 3 \times 4 \times \ldots \times N$$

Combinations are an interesting calculation in poker, and illustrate a potential use of the factorial operator. To calculate the number of five-card combinations in a 52-card deck, we use the combinations formula:

$$\text{Possible poker hands} = {}_{52}C_5 = \frac{52!}{5! \cdot (52-5)!}$$

Factorial calculations of even small integers grow rapidly, requiring the use of the java.math.BigInteger class. Calculations of factorials result in a large number of integer multiplications.

The factorial calculations shown in Figure 6 do show some consistent, limited benefit from Hyper-Threading. Indeed, for four threads the IBM JVM shows a 17% increase in performance using an HT-enabled system.

Incidentally, there are 2,598,960 five-card combinations in a 52-card deck.

Test 3: 150K calculations of Math.tan() (Floating point, mixed stack)

This test simply calculates the tangent of an angle 150,000 times in a tight loop (see Figure 7).

All Java threads have two call stacks: one for Java calls, the other for C calls. The java.lang.Math.tan(double) function is native, calculating an approximation of tangent with a 27th order polynomial. It's likely that the reason this operation scales so well into Hyper-Threading is the constant call stack switching, giving the processor time to utilize its secondary thread context.

www.LinuxWorld.com

The Leading Magazine for Enterprise and IT Management

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 necessary 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* does not feature low-level code snippets but focuses 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 presents a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features 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

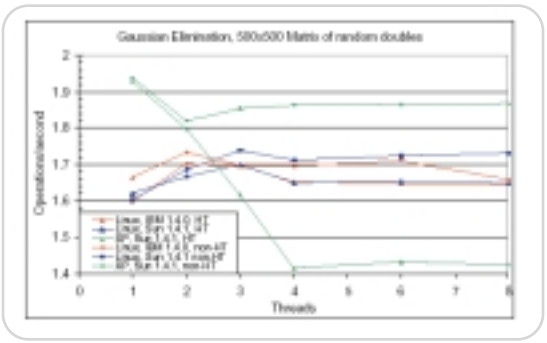


Figure 5 Gaussian elinination test

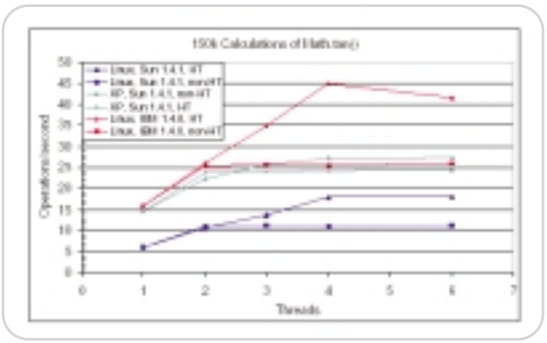


Figure 7 Target calculation test

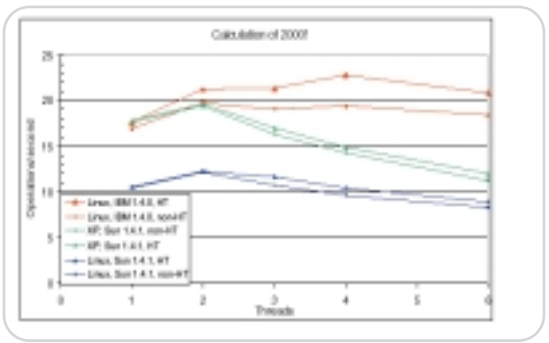


Figure 6 Factorial test

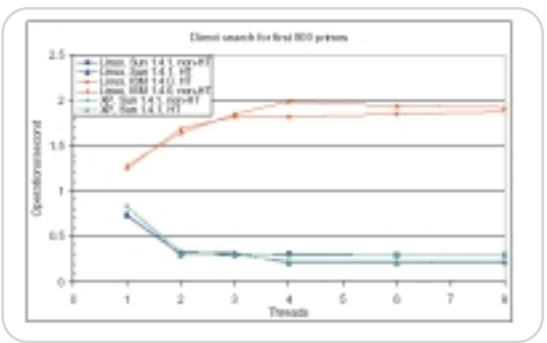


Figure 8 Prime number search test

Test 4: Prime number search

A prime number search operation was created using the BigInteger class and a very simplistic direct search factorization. The poor algorithm is not as important as the type of calculations being performed. This class performs a large number of BigInteger divisions.

It is difficult to tell what is going on in Figure 8, beyond the fact that the IBM JVM is beating Sun's. The IBM JVM scales well into threading this operation. It does even better when Hyper-Threading is enabled. The Sun VM scales poorly into threads, and it becomes worse with additional thread contexts. You could speculate that this behavior is characteristic of a low-level synchronization contention issue in the Sun JVM.

Testing Summary

The plots above give some general idea of how these various operations scale into threads. In most cases, the HT performance gains are modest. The following is a summary of performance differences seen with Hyper-Threading enabled versus disabled for each of the tested JVMs.

IBM 1.4.0, Linux 2.4.18

Threads	*Gauss*	*Factorial*	*Math.tan()*	*Prime*
1	4.13%	3.92%	-0.10%	3.06%
2	1.92%	7.39%	1.62%	-2.42%
3	0.21%	11.45%	34.99%	1.96%
4	-2.58%	16.98%	75.84%	9.84%
6	-3.56%	13.33%	60.96%	4.53%
8	-0.69%			2.41%

Sun 1.4.1, Linux 2.4.18

Threads	*Gauss*	*Factorial*	*Math.tan()*	*Prime*
1	0.99%	0.28%	-0.75%	0.30%
2	-1.20%	0.35%	-1.76%	6.10%
3	-2.20%	8.21%	23.76%	6.30%
4	-3.63%	8.28%	62.74%	-30.08%
6	-4.13%	7.71%	62.96%	-27.50%
8	-4.73%			-28.28%

Sun 1.4.1, Windows XP Pro

Threads	*Gauss*	*Factorial*	*Math.tan()*	*Prime*
1	-0.51%	0.93%	0.62%	-1.32%
2	-1.18%	0.98%	-6.17%	14.07%
3	-12.90%	3.53%	7.85%	-0.74%
4	-23.96%	4.61%	11.74%	-24.14%
6	-23.23%	6.35%	11.79%	-23.46%
8	-23.66%			-23.36%

Conclusion

When I began this project, I fully expected to see marked performance gains using Hyper-Threading over identical hardware not using HT. In the course of testing, I've learned quite a bit about performance differences for Java on various platforms, hardware configurations, and virtual machines. Hyper-Threading is not the boon I had expected. In some situations, performance gains for HT reached the 75% mark, which is considerable. There was little significant performance degradation using HT, so using it seems to be largely on the upside.

Perhaps the more important finding is that the IBM JVMs perform significantly better than the Sun JVMs. In addition, the IBM JVMs scaled far better with threads than did Sun's offering. If performance is of key concern, and you're not using some of the more esoteric features of the Sun JVM, IBM JVMs deserve serious consideration.

Most server-side Java applications are not doing computationally intensive tasks. The tasks focus more heavily on socket IO – communicating with databases, clients via HTTP, RMI, Web services, and the like. Processors will be given plenty of socket IO wait time to schedule parallel tasks. For socket-IO-bound applications, be sure to consider the relative skill of your operating system in the IP arena.

The introduction of Hyper-Threading on desktop P4 systems is also exciting. Java developers often develop on Windows or Linux-based desktop systems and deploy onto larger SMP and potentially SMT systems. HT will allow a desktop developer and user to see some of the benefits of

Paul Bemowski is an independent consultant, focusing on Java solutions to enterprise computing problems.

bemowski@yahoo.com

threaded applications long before deployment to the higher-end systems.

SMT technology is here to stay. Intel's Hyper-Threading implementation is sure to be the first of many. Chip industry watchers speculate that Simultaneous Multithreading and thread-level parallelism will spell the ultimate end of the "megahertz wars." A chip's performance will be tied less to its internal clock speed and more to the bells and whistles it incorporates. Other chip manufacturers are sure to follow suit, and all implementations will improve in quality over time.

Operating systems are also continually improving their support for Hyper-Threading. It does seem strange that the performance on an XP system, which should be HT optimized, was often less HT friendly than the 2.4.18 Linux kernel, which is HT ignorant. As more sophisticated support for HT is built into operating systems, we should see more significant performance gains using HT in the Java world.

The combination of Java and Linux in the datacenter is rapidly gaining ground on the Solaris/Java platform. The majority of these new Linux servers are running high-end Intel-based hardware. Hyper-Threading will give this trend a further push in the Linux direction.

For now, given a piece of hardware that's HT capable, the configuration that offers the best performance under most conditions is the IBM 1.4.0 JVM on Linux with Hyper-Threading enabled.

Resources

- *Microsoft license clarification for SMT systems:* www.microsoft.com/nz/licensing/downloads/hyper_threading_processors_licens ing_brief.doc

Intel Processor Specsheets

- *Xeon:* www.intel.com/products/server/processors/server/xeon/index.htm
- *Xeon DP:* www.intel.com/design/xeon/prodbref/index.htm
- *Xeon MP:* www.intel.com/products/server/processors/server/xeon_mp/index.htm
- *Pentium 4:* http://developer.intel.com/design/pentium4/datashts/298643.htm
- *P4 Chipset matrix indicating HT support:* www.intel.com/design/chipsets/linecard.htm
- *IBM Whitepaper on Linux and Hyper-Threading:* www-106.ibm.com/developerworks/linux/library/l-htl/?dwzone=linux
- *LinuxWorld article indicating Q4 2003 release of 2.6 Kernel:* www.inuxworld.com/story/33805.htm

Glossary

- ***Physical processor:*** A silicon-based hardware processor
- ***Logical processor:*** A hardware/software system making pseudo-parallel use of a single physical processor
- ***Simultaneous Multithreading (SMT):*** The use of logical processors to increase processing throughput on a single physical processor
- ***Symmetric Multiprocessing (SMP):*** The use of multiple physical processors in parallel, each running separate threads of execution
- ***Hyper-Threading:*** Intel's marketing name for its SMT technology on Xeon and Pentium 4 processors

Glen Cordrey
J2ME Editor

Sun **Addled**

I'm just back from vacation, and after six days of sun on the beach in the morning and on the tennis court in the afternoon, sun addled is a good description of my frame of mind. Also account for liberal quantities of beer throughout the week, and I'll be happy if I'm somewhat coherent in this editorial. I'm also in a post-vacation funk, so forgive me if I wax philosophical while I strive for coherency.

The vacation's one of the reasons I didn't go to JavaOne this year – it would have been hard to justify two weeks off so close together, and this time play won out over work. On some occasions work and play have mixed without undue harm to either – I've written J2ME code while on a ski vacation in Tahoe and on vacation in the Caribbean – but that was due to extraordinary circumstances (the rollout of the first J2ME cellphones in the U.S.). This vacation was completely disconnected from the Internet – I intentionally left my laptop home, and my cell phone was for emergency calls only. I'm a bit of a Luddite in that I don't want to be connected and available 24x7, and don't spend large amounts of my free time playing with the latest gee-whiz gadgetry. There are a lot of other things to do in life, and you only get one shot at it (reincarnation notwithstanding).

I think that the rate of technical change in our industry puts developers at a real risk of early burnout. Alvin Toffler wrote that "Future shock [is] the shattering stress and disorientation that we induce in individuals by subjecting them to too much change in too short a time." As a Java developer you may sometimes feel that you're experiencing such future shock. If you've been in the Java business for more than a few years, consider the changes that you've seen in that time. New APIs proliferate like rabbits – XML, JAXB, SOAP, CLDC, MIDP, JDK 1.4, the list goes on and on. At 992 pages, the current issue of *Java in a Nutshell* is over twice the size of the first edition. This nutshell has grown from a hazelnut to a coconut, and in a few years I expect to have one bookshelf devoted just to all of the editions of this book that I've acquired (yes, I could throw out the old editions, but then I'd have to transfer all of the highlighted items and dog-ears from the old editions to the current one).

If you don't keep abreast of the latest changes by reading technical journals and making frequent book purchases you can quickly find your skills becoming, if not obsolete, certainly in far less demand. If you're content to specialize in a single area – say, JSPs or EJBs – you can save yourself some of this, but then you limit yourself career-wise, and you won't experience the satisfaction of architecting larger solutions.

Of course, change is part of what keeps our work interesting, and keeping abreast of these changes helps give you the hot skills that bring in the bucks. But trying to keep up with what's coming out from Sun can leave you Sun addled. Add in a desire to keep current in object-oriented analysis and design, UML, design patterns, and the full complement of related technologies, plus the pervasive "need it yesterday" mentality of modern organizations, and you can easily feel overwhelmed.

You really can't expect to know everything about everything in the technologies we deal with, so know when to go full bore and when to throttle back. As The Specials (and many before them) sang, "enjoy yourself, it's later than you think." If the alternative is getting so burnt out that you end up considering a career change, wouldn't you rather chill out every now and then by sipping a Corona on a beach somewhere?

• • •

In last month's editorial I surveyed the MIDlet marketplace. Fortuitously, next month's issue of ***JDJ*** will contain an article by Greg Schwartz that takes you through the process of actually getting your MIDlet published via various outlets. So if you've thought about selling MIDlets, keep an eye out for this article.

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

Sun Addled

I think that the rate of technical change in our industry puts developers at a real risk of early burnout. Alvin Toffler wrote that "Future shock [is] the shattering stress and disorientation that we induce in individuals by subjecting them to too much change in too short a time." As a Java developer you may sometimes feel that you're experiencing such future shock.

46

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

JDJ Workshop with Alan Williamson

M	T	W	T	F	S	S

Coming to you...

May:	June:
NEW YORK	ATLANTA
DALLAS	BOSTON

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 EDUCATION

PRACTICAL INTEGRATION OF JAVA-ORIENTED TECHNOLOGIES AND 2.5/3G PLATFORMS

Challenges for the wireless developer

by Eva Skoglund, Angus McIntyre, Bryan Hartlen, and Marc R. Erickson

Broad ranges of deployment choices are available to developers as they engage in new projects that will leverage the power of wireless communication platforms. While deploying services through third-generation (3G) and second-and-a-half generation (2.5G) wireless terminals presents challenges that are quite different from those encountered in personal computers 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 what you need to know when dealing with resource constraints and the management of software components through development, testing, deployment, and maintenance.

Communication handsets already offer functions similar to those available on desktop computers, including Internet access. While handheld devices are becoming more capable and powerful with each generation of products, they continue to be resource constrained compared to their desktop brethren. Limited amounts of memory, processing, and battery power are realities for the wireless developer.

These hardware limitations must be compensated for by smart application development and tight middleware integration. Rather than just acting as browsers for reformatted Web content, 3G handsets have the potential to host application software specifically built to implement client and service aspects of a solution. Since developers are no longer restricted to "thin-client" approaches, components installed directly within wireless devices have greater access to device features, sensors, and actuators. Clearly, developers will need access to a new generation of end-to-end tools that work within an open, universal tools integration platform. These will help address the special needs of multi-language, multi-platform, and multi-vendor development environments.

The challenge is to combine flexibility and broad functionality with the demands for very limited memory, processing, and battery power. 3G handsets will be built on a component base that provides the robust and reliable platform on which applications can be hosted. 3G is where the phone world meets the computing world. Voice communication will continue to be central while multimedia, secure e-business transactions, and data services will provide differentiation. A 3G handset will need to have the same reliability as a phone, but the same functionality as a computer, and still be small and power efficient.

Let's use an example to illustrate the reality of merging voice, data, and communications onto modern-day devices (cellphones, PDAs, and handheld computers) and enabling a new dynamic e-business application model. A technician on a service call uses a handset to view a full-motion video on how to remove a part from a photocopier. Using the same handset, the technician submits an order for the part and receives verification that the part has been reserved for his customer. Using location-based services, the server application, determining that there is a delivery van with the part on board in the neighborhood, uses a cellular SMS message to divert the driver to the technician's location.

As applications are extended to mobile devices, new technologies will be required to provide productivity to the end users while shielding them from the inherent complexities of integrating local and remote services. Local services such as bar code scanning, scalable vector graphics, full-motion video, and voice navigation will be required on the device. Remote services residing within the service infrastructure, such as location-based (GPS), transactional, and messaging services, will need to be accessed wirelessly by the device.

All these technologies exist today. To successfully integrate and deploy them on a constrained handset is a "simple matter of programming." To get there faster requires two things: the right platform and the right tools.

The role of Java in these systems is hidden from the end user, but it will play an important role for the developers of the wireless terminals and infrastructure equipment.

Next Generation Communication Systems

What are the reasons for using Java in 3G/2.5G terminals and infrastructure? Java will be successful within the

communication segment for the same reasons it's successful in the IT segment: (1) time-to-market through productivity, and (2) flexibility and stability through open standards.

Shifting market demographics, global competition, and significant development and maintenance costs have reinforced the need for increased flexibility and productivity. No one application or service is going to meet all needs, and as a result service providers must have the ability to quickly and efficiently deploy new applications. This drives the fundamental idea of building a general platform with predefined capabilities and shared services that can be configured in real time with applications and add-ons, resulting in differentiable end products. The J2ME specifications provide a definition of such a platform for devices of varying capabilities. Developing applications according to these J2ME specifications:

- Ensures application compatibility and portability with device capabilities
- Reaps the productivity benefits of the Java programming model and accessibility to a continually growing base of skilled developers
- Simplifies the integration of applications and services that reside in both the handset and infrastructure sides of the wireless link

Selecting the best J2ME implementation is important, as it's the implementation of the "platform" – the underlying real-time operating system (RTOS) and virtual machine – that provides to a varying degree:

- A safe and secure programming environment that protects the system from faulty add-ons that could corrupt the system
- A layered software design with isolation mechanisms that prevent direct access to hardware and RTOS services
- A fast and reliable execution environment with minimal memory consumption and efficient power usage that enables applications to take full advantage of available hardware resources in order to produce high quality service
- Tested and certified, not just compatible, implementation against the J2ME specifications' "Java-powered" runtime environment
- An extensible platform for connecting applications deployed on the handset with data and applications within the infrastructure
- Runtime downloading (and unloading) capabilities of both Java applications and native system components
- Efficient and intuitive programming tools for all of the above

In the following section, OSE Systems and IBM describe how the integration of RTOS and J2ME-certified runtimes can be optimized for the wireless handset market. Experience is taken from their recent efforts in integrating OSE Systems' RTOS OSE and IBM's WebSphere Micro Environment.

Development took place in an integrated tools platform with development tools based upon the open source Eclipse platform, an open universal tools integration environment. Development tools that are powered by Eclipse technology such asIBM's WebSphere Device Developer (an offering specifically targeted to embedded platforms) are also discussed as examples of the tooling needed for embedded Java applications.

The Right Platform – Not All J2ME Implementations Are Created Equal

J2ME is a direct response to the recognition that not all devices are created equal. It provides a means for defining various device classes and a minimum level of capability and services that can be guaranteed on compliant devices. It does not, however, specify how well those services will perform. The devil is in the details of the platform's design, functionality, and implementation – how the individual components of the platform, the RTOS, and the virtual machine are designed and implemented, what capabilities they have, and collectively how they are integrated.

General Performance

Footprint and Memory Costs

One of the largest costs in a 3G handset is the memory. Designing a device with loads of expensive RAM and high-performance flash is not an option for many manufacturers. The right platform will be based on software specifically designed from the ground up to be small, scalable, and configurable, not carved down from an enterprise implementation.

The RTOS plays an important role in saving memory, not only by providing a scalable, small footprint solution, but also by implementing memory-conserving features such as:

- Limiting the amount of RAM consumed during execution
- Preventing memory fragmentation
- Supporting persistent memory types and enabling flash memory to function as the work memory, including "instant-on" execute-in-place (XIP) features that allow platforms to execute code directly from the flash memory without prior loading into a RAM buffer
- Supporting dynamic remapping of applications' memory pages between RAM and flash

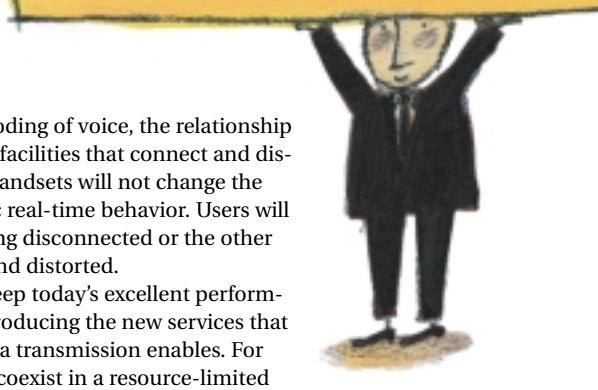
Building on a strong RTOS, J2ME platforms at the simplest level assist in reducing the memory footprint for small devices by specifying the set of services available to an application on a compliant platform. This allows these services to be implemented once and then be shared by each application. In the wireless domain this also means improved performance as only the application, not the core services, need to be deployed over the wireless connection.

At a deeper level, developers also need to look at the overall memory of an application as it runs on the device. Some virtual machines grab small amounts of memory up front (to reduce the initial footprint), and incrementally grab memory as it's needed (costing in terms of speed). Others grab more memory up front, but less over time (up front memory hit, but faster over time). The ideal solution is a virtual machine that is customizable and tunable. With J9, the virtual machine inside WebSphere Micro Environment, initial and incremental memory heap space can be set to meet the specific service needs and device capabilities.

Finally, once the application has been designed and optimized, there are benefits to be realized in how it is deployed to the device. Depending on the type of application and the characteristics of the device, it may be more beneficial to execute the application out of flash rather than RAM. This could be to minimize RAM requirements or to provide the application with an "instant-on" capability.

Execution

The execution speed of applications on these consumer handsets will continue to be an important success factor. Java applications have previously suffered a bad reputation



because of their interpretive nature. While it's true that interpreters will generally run slower than compiled applications, the difference at the application level is not necessarily noticeable; in situations where it is noticeable, there are optimized runtime options such as ahead-of-time (AOT) and just-in-time (JIT) compilation.

Just-in-time (JIT) compilation is a runtime option that uses a compiler within the virtual machine to translate Java bytecodes into native machine language for a specific device. A cache of memory contains compiled machine language prepared "just in time" for use. Execution of recurring bytecodes is then handled directly from the machine language in cache. After the initial "compilations," JIT technology improves performance at the cost of increased memory and start-up time.

Ahead-of-time (AOT) compilation is an option that converts Java bytecodes into native machine language for the specific device at deployment time. Unlike JIT, compilation occurs on the development host. The result can be packaged "ahead of time" into a compact executable for direct loading and execution within the virtual machine environment. AOT compilation can be extremely effective when applied selectively to application hotspots. It can provide near JIT performance without the start-up time penalty and at a fraction of the memory costs.

The J9 virtual machine was designed first and foremost to interpret bytecodes quickly. Building on this strong foundation, the JIT and AOT compilers provide the application developer with options for managing the speed versus memory cost trade-off. The significance of this flexibility is considerable given that the J9 virtual machine can execute applications deployed as bytecodes, AOT or JIT compiled machine code, and, most important, any combination of the three.

Application Security and Continuity

While ensuring reliability of standalone software applications, it's equally important to design a device with system robustness in mind. Downloaded applications and services cannot be allowed to corrupt the system. Separation of system-critical and noncritical software is essential and the operating system in the device should provide isolation mechanisms and memory protection for this purpose. For example, in the case of OSE's dynamic loading/unloading capabilities of native software (C/C++), special protection mechanisms have been introduced to ensure that loaded modules will be memory protected when taking advantage of a hardware MMU.

Introducing a Java runtime environment that takes advantage of the RTOS' inherent security in a 3G handset has the great benefit of providing a secure and protected environment for application software, which is often written in Java. The handset's native environment (C/C++) that controls the device will be protected from the Java application environment. In system architectures that would benefit from greater separation, multiple virtual machines could exist on the handset and can be started and stopped independent of each other. In addition, when the RTOS supports native runtime downloads, even new, updated Java Virtual Machines can be dynamically installed after a handset is produced and deployed.

Deterministic Real-Time Behavior

Deterministic real-time behavior is a requirement in today's 2G mobile phones. To wirelessly deliver your voice to any location in the world within fractions of a second, 2G platforms include software that must control radio components, digital coding and decoding of voice, the relationship with base stations, and setup facilities that connect and disconnect calls. Moving to 3G handsets will not change the requirement for deterministic real-time behavior. Users will not accept calls suddenly being disconnected or the other party's voice sounding slow and distorted.

The challenge will be to keep today's excellent performance and reliability, while introducing the new services that high-speed, packet-based data transmission enables. For multiple Java applications to coexist in a resource-limited device, the Java runtime environment needs deterministic real-time behavior.

The design of the virtual machines has also moved forward to allow for deterministic execution. One specific area is in the garbage collection of the virtual machine. To illustrate, the IBM J9 virtual machine inside WebSphere Micro Environment has the following features to assist in the development of deterministic applications.

- Real-time systems need to recover memory efficiently. Look for features such as precise garbage collection (GC) as opposed to conservative garbage collection. GC should also be incremental, so the virtual machine is not locked during GC.
- In the most serious of deterministic cases, GC should also allow interrupts. This stops GC immediately so critical events can be processed at the highest possible priority. An example of this would be receiving an incoming phone call while gaming.

Having a virtual machine implement deterministic functions has limited value if the underlying RTOS does not predictably implement real-time elements. Operating systems such as OSE are an example of a true embedded real-time system, designed from the ground up for demanding needs like determinisms, reliability, and robustness. Besides the classical features such as preemptive scheduling, lightweight processes, small footprint (measured in tens of kilobytes), scalability, efficient interrupt handling with short interrupt latency, and an interrupt-centered device driver model, there are other features to look for in an RTOS:

- Message-based architecture to ensure ease-of-use, high performance, advanced debug capabilities, transparent IPC in distributed systems, and memory-protected systems
- Memory management with a focus on performance even when using low-performance memory types, efficient prevention of memory fragmentation, and multiple memory pools to allow robust system design
- Automatic error handling with a multilayered error handler approach
- Process supervision and automatic resource cleaning, which enables a system to be self-recoverable

The Right Tools – The Java Developer Is Efficient

As the 3G handset becomes a deployment platform within an overall pervasive computing context, tools to manage the end-to-end project life cycle will be needed. It will be customary to establish new development relationships as part of these projects. Often several companies or departments will need to collaborate on solutions that deploy across deeply embedded controllers, handsets, personal computers, servers, and even mainframe systems that handle millions of devices concurrently. The development life cycle starts with managed requirements, then progresses to modeling, code development, debugging, installation and management tasks, and testing. Testing takes on new aspects as

multiple handset platforms, deployment networks, Web services servers, and service providers must be accommodated. Tools that manage all of these aspects of project development and deployment must integrate together well if developers are to progress efficiently.

As in the device runtime discussion, it will be important for the Java and native (C/C++ and assembler) tooling environments to be well integrated. Being able to switch between task-aware freeze-mode and run-mode debugging is important in real-time systems as certain bugs cannot be found unless parts of the system can execute while others are frozen.

Traditional source code debugging needs its complement in system-level debugging where events can be traced and analyzed, and design flaws can be found. Advanced system-level debugging is possible when the system design is based on a message-passing architecture with automatic resource cleaning, like OSE enables. Profiling of the CPU and memory usage as well as postmortem debugging are other needed debugging tools. The OSE Illuminator debug tool suite is implemented in Java, which provides for easy integration with a Java development environment like Eclipse-based WebSphere Studio Device Developer.

While Java's portability and productivity benefits are well known, tooling designed for the embedded developer will need to provide several features above and beyond those found in a simple IDE. Real productivity is achieved with the following:

- The ability to develop and debug in both on-host emulation environments and the actual on-device environment
- The ability to develop and debug mixed mode, Java, and C/C++ applications
- Execution analysis tools that operate on the device and highlight application bottlenecks by profiling memory, stack and thread usage, and priorities
- Setup and configuration wizards that ensure applications are developed against the desired J2ME configuration or profile and for the desired target hardware
- Wizards for recommending which portions of the application should be compiled as Java bytecodes, AOT, or JIT executables
- Wizards that package an application for execution out of flash memory (XIP)
- Wizards for packaging applications for deployment within an OSGi or enterprise side service
- Ability to seamlessly create new custom tool features or to integrate new or better tool capabilities from third-party providers

There are a large number of competing and complementary operating platforms and middleware technology offerings available. New project and business relationships may mean that a developer needs to bridge between multiple tools and collaboration environments. Since projects will often need to address additional deployment environments, including creating and managing metadata for complex middleware, multiple compatible development offerings may need to be considered. When an open, extendable integration framework like Eclipse is the basis for tools, it becomes easier to accommodate multiple-platform, multiple-language, and multiple-vendor development tasks.

Part of the challenge for developers working to deploy solutions across server and wireless handset platforms involves collaboration and team coordination. A universal tools integration platform like Eclipse is equally at home in supporting a variety of development languages (Java, AspecJ, C/C++, Cobol, and Smalltalk, for example) and a broad range of enterprise and embedded deployment platforms. Eclipse supports all these environments through common access to modeling tools, testing environments, and concurrent access to multiple source code repositories. Several vendors are now offering solutions that integrate their areas of specialization with the Eclipse platform.

Offerings like QNX Momentix and TimeSys TimeStorm use the same Eclipse platform and provide speciality tools for working with QNX Neutrino and TimeSys Linux in C and C++ language environments. Source code control and management offerings from Computer Associates (All Fusion), MERANT (PVCS), Rational (XDE), Borland (Together Control Center), MKS (Source Integrity), Serena (Workbench), LogicLibrary (Logidex Asset Expert), Eclipse (CVS), Starbase (StarTeam), Telelogic, and Instantiations (CodePro Studio) let you plug in access to a variety of VCM and source code management systems. Offerings like IBM's WebSphere Studio Application Developer extend Eclipse to include visual editors, development of enterprise services, middleware access, and more. Organizations like Quest Software, Parasoft, JUnit.org, and Scapa Technologies have provided test extensions.

Summary

3G handsets will need to combine phone reliability and real-time requirements on small and resource-constrained devices with ever-changing application functionality. J2ME specifications provide definitions of general platforms for devices of varying capabilities and will serve as the basic software building blocks for these devices. Successful implementations of the J2ME platform will require an RTOS and Java runtime environment that focus on deterministic real-time behavior and low memory usage. Successful projects will need to efficiently integrate and work with tools from multiple vendors for multiple development and deployment platforms and in multiple languages. The combination of J2ME technology and Eclipse-based tools is the key to powerful solutions.

Eva Skoglund is a product marketing manager for wireless systems for OSE Systems, a real-time operating systems company. She holds an MS in computer science from the Royal Institute of Technology in Stockholm.

Bryan Hartlen works in business development, product marketing, and IP licensing areas and manages IBM's family of embedded Java products. Previously, he developed or managed software developers on platforms that ranged from mainframe applications to shop floor control systems to PCMCIA cards.

Marc Erickson has been a key participant in several international innovations from IBM including workflow technology and service support center development. He is now on assignment as the communications manager for Eclipse.

Angus McIntyre is the director of business development for embedded technology at IBM and has spent his entire career on application development tools (OTI, IBM Canada Lab, VisualAge for Java, VisualAge Micro Edition). He holds a BS in computer science and statistics and an MBA.

The Spring 2003 Rocky Mountain Software Symposium

'No Fluff, Just Stuff'

Tom Marrs

Because today's software development climate is so fiercely competitive, only those developers, architects, and technical managers who meet the needs of their clients and embrace change can hope to survive. The "No Fluff, Just Stuff" (NFJS) conference series (www.nofluffjuststuff.com) covers the latest in industry developments and best practices with cutting-edge sessions and panel discussions on Java/J2EE, XML, open source, and agile methodologies. In this review, I discuss my experiences at the recent Spring 2003 Rocky Mountain Software Symposium (RMSS), held from Friday, May 16, through Sunday, May 18, in Denver.

The speaker list for the Spring 2003 RMSS reads like a "who's who" in the Java/J2EE/open source community. For the complete lineup, please visit www.nofluffjuststuff.com/2003-05-denver/speakers.jsp. I attended sessions given by:

- ***Sue Spielman:*** Author of *The Struts Framework: Practical Guide for Programmers* and the forthcoming *JSTL: Practical Guide for JSP Programmers*
- ***James Duncan Davidson:*** Creator of Jakarta Tomcat and Apache Ant
- ***Bruce Tate:*** Author of *Bitter Java* and lead author of *Bitter EJB*
- ***Dave Thomas:*** Author of *The Pragmatic Programmer: From Journeyman to Master*
- ***Erik Hatcher:*** Author of *Java Development with Ant*
- ***Maciej Zawadzki:*** Creator of the AntHill build management server and author of Urbancode's EJB Benchmark
- ***Ted Neward:*** Author of *Server-Based Java Programming* and the forthcoming *Effective Enterprise Java*
- ***Glenn Vanderburg:*** Principal with Delphi Consultants

One of the highlights of the conference was the keynote – "Introduction to Pragmatic Programming" by Dave Thomas. This session showed the inefficiencies in today's software development practices and what could be done about them. We experience the same problems today that we had back in the 1970s and '80s. Projects are still late, underfeatured, over budget, and full of bugs. Dave cited an NIST study that showed how the U.S. wastes $60 billion each year on buggy software. Since there are about 3–4 million developers in this country, each developer costs the U.S. economy roughly $20K each year – ouch!

Pragmatic programming fits somewhere between methodologies and technology-specific skills. It is not a methodology; it's about the individuals and teams who produce software. Pragmatic programming breaks down into three practice areas: individual, technical, and team.

The most distinctive feature was the section on individual practices:

- ***Learn continuously:*** Read books, go to conferences, and interact with other developers at user groups.
- ***Fix broken windows:*** Fix problems as soon as they occur to keep them from growing larger.

Speaker Interaction

The speakers were personable and helpful. After presenting on "Effective Enterprise Java: Systems," Ted Neward spent a long time with me discussing ClassLoaders, their relationship to the JVM, and how this relates to Singletons in a J2EE environment. In between sessions, Sue Spielman spoke with me about the ups and downs of writing a technical book, and compared being a solo author to working with other authors. During breakfast, Bruce Tate gave me ideas on how to become more marketable in today's challenging IT economy. After one of his JAAS presentations, Maciej Zawadzki provided the inside scoop on Urbancode's EJB Benchmark – he created a series of benchmarks to prove to his clients that Entity EJBs were needlessly slow, and that alternate solutions would perform and scale better.

Conclusion

When I asked attendees for their opinions of the conference, the consensus was:

- They gained valuable insights from the sessions that they could apply to their daily work.
- Their time wasn't wasted with vendor product demos.
- They enjoyed interacting with recognized industry experts.
- The conference schedule over the weekend didn't interrupt their schedules.
- It was less expensive and easier to attend a local conference than having to travel.
- The conference was a good value for the price.
- The conference CD contained all the presentations.

Since I learned so much in just one weekend, I would recommend an NFJS conference to any developer, architect, or technical manager looking to keep up with the latest trends and improve their skills. I plan to attend this conference when it returns to Denver in the fall because the NFJS symposium series truly lives up to its billing – "No Fluff, Just Stuff."

Tom Marrs is the principal/senior software architect at Distributed Computing Solutions, Inc., based in Denver. His focus is on building J2EE system architectures and mentoring developers in J2EE, OO, XML, and Web services technologies.

thomasamarrs@comcast.net

Industry **News**

Aplix, Rococo Partner for Java-Bluetooth Platform

(*London*) – Java specialist Aplix is joining with Bluetooth vendor Rococo Software for an integrated Java-Bluetooth platform targeted at mobile phone OEMs and integrators.

The alliance will enable mobile phone manufacturers to implement a single, integrated solution for Java-Bluetooth technology support in Aplix's JBlend Java Virtual Machine (JVM) environment. The companies will jointly market and sell the platform.

www.aplixcorp.com
www.rococosoft.com

Unisys Introduces Mainframe-Scale Java Solution for Windows

(*Blue Bell, PA*) – Unisys Corporation has announced the availability of Java Virtual Machine (JVM) for ES7000 servers, a Java computer platform that takes advantage of Microsoft's operating environments. Now enterprise solutions created with Java software technology can tackle workloads that once could be handled only by expensive Unix-based servers by applying the 32-processor scalability of the Unisys ES7000. This new Unisys JVM can be downloaded from the Internet free of charge.

www.unisys.com

Strategic Focus Compares Developer Productivity Using Business Process Management Software and Java IDE

(*Milpitas, CA*) – Strategic Focus, a business strategy and software evaluation consulting firm, has completed a study that compares the developer productivity of building applications with business process management (BPM) software and a traditional Java development environment. Strategic Focus built a BPM application in each of the two development environments (BPM and Java IDE) following identical specifications, then measured the time needed to complete each phase of the software life cycle. A free copy of the 17-page white paper can be downloaded from www.pega.com or www.strategicfocus.com.

Patent-Pending Type 4 JDBC Drivers Providing Single Sign-on

(*Rockville, MD*) – DataDirect Technologies, a provider of components for connecting data and applications, has announced DataDirect Connect for JDBC 3.3, the first set of Type 4 JDBC drivers to support Windows authentication. DataDirect Technologies has implemented Windows authentication within their Type 4 JDBC components used for connecting Java applications to Microsoft SQL Server data. DataDirect Technologies has a patent pending on this new technology.

www.datadirect-technologies.com

Panscopic Application Version 2.3 Released

(*San Francisco*) – Panscopic, a provider of J2EE self-serve reporting, has announced the availability of version 2.3 of the Panscopic Application. New features include XML data access and transformation capabilities via integration with BEA's Liquid Data, the ability to build ad hoc reports using any existing query or report, and integration with Microsoft Excel, enabling enterprise data to be accessed from native Excel spreadsheets.

www.panscopic.com

Wily Integrates Introscope with IBM WebSphere Application Server 5.0

(*Brisbane, CA*) – Wily Technology has announced the direct integration of Introscope, Wily's Enterprise Java Application Management solution with IBM WebSphere Application Server 5.0 for z/OS and OS/390. Introscope assists enterprises in the deployment and management of mainframe Java applications that meet the highest demands for performance and reliability.

www.wilytech.com

NewMonics Wins Navy Contract for Hard Real-Time Java

(*Tuscon, AZ*) – NewMonics, Inc., was awarded a U.S. Navy contract to implement the integration of hard and soft real-time Java components for the large software systems targeted by the Navy's Open Architecture Computing Environment (NOACE). The purpose of Navy Open Architecture (NOA) is to evolve Navy surface ship warfighting systems from the current status quo toward a unified Navy product line composed on a common computing environment (e.g., NOACE), a common set of warfighting functions shared across many platforms, and unique sets of warfighting functions associated with each ship class.

www.newmonics.com

Thor Technologies Teams with Oracle

(*San Francisco*) – Thor Technologies, Inc., a provider of secure enterprise provisioning and an active member of the OraclePartnerNetwork, has announced the immediate availability of new Xellerate adapters for Oracle Internet Directory, the identity repository for the Oracle9*i* platform. The combination of these technologies gives organizations a comprehensive solution for effectively managing user identities and access rights to heterogeneous enterprise systems and applications. The result for enterprises is streamlined identity management capabilities and reduced administration. Thor is also a featured Oracle9*i* Application Server partner.

www.thortech.com

BEA WebLogic Platform 8.1 Ships

(*San Jose, CA*) – BEA Systems, Inc., has announced the general availability of BEA WebLogic Platform 8.1, WebLogic Workshop 8.1, WebLogic Integration 8.1, and WebLogic Portal 8.1. BEA WebLogic Platform 8.1 and each individual product, including the previously available WebLogic Server 8.1 and WebLogic JRockit 8.1, can be downloaded now from www.bea.com, or customers can receive the development versions of the products at no charge for one year as part of the BEA dev2dev subscription program. BEA's new products are based on a unique software architecture developed by BEA, all based on Java 2 Enterprise Edition (J2EE), extensible mark-up language (XML), and emerging Web services standards.

Caché 5

by InterSystems Corporation

Reviewed by
Mathias Kühn &
Christian Och

InterSystems Corporation

Cambridge, MA 02142 USA
Web: www.intersystems.com
Phone: 617-621-0600
E-mail: webmaster@intersys.com

Specifications

Platforms: Windows 95/98/ME, Windows 2000 Pro/2000 Server/2000 Adv. Server, Windows XP Home/Pro, Mac OS X, Linux, HP/UX, Solaris, IBM AIX, Tru64, OpenVMS
Pricing: $200 in a single-user version and starting at $1,000 for a multi-user configuration

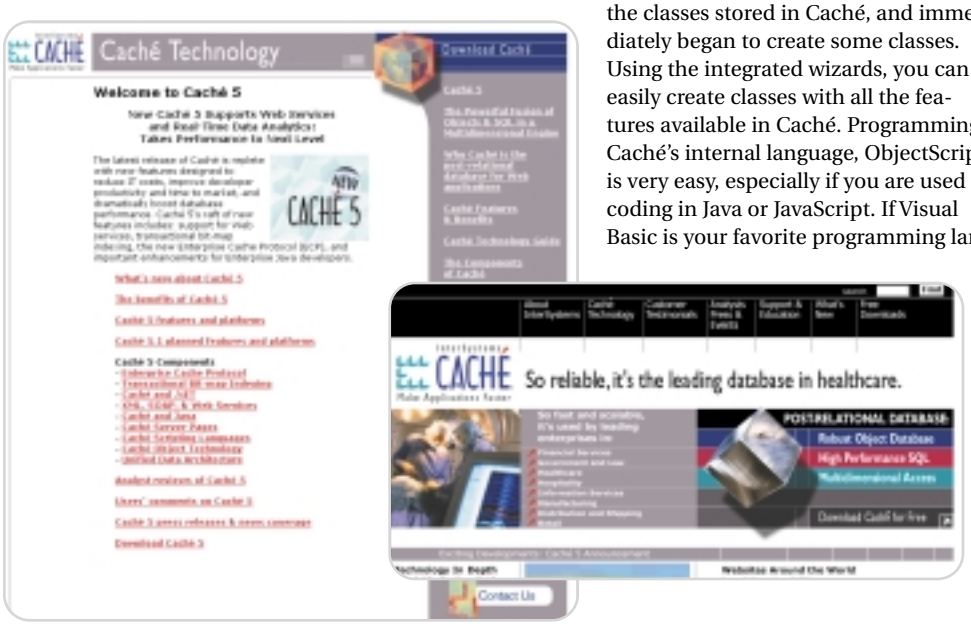
Test Environment

Notebook ASUS L3800 with Mobile Intel Pentium P4 processor, 40GB disk, 1GB RAM
Operating Systems: Windows 2000 Pro with Service Pack 3/SuSE Linux 8.1

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.

We set out to test the capabilities of this database software from a Java developer's perspective. Working with the latest release of the software, Caché 5, we found the results impressive.

Caché's unique combination of objects and SQL provides a very interesting development framework for object-oriented programmers that goes far beyond what pure relational or pure object-oriented databases provide. Besides eliminating the impedance mismatch between objects and tables, Caché's unified data architecture allows rapid application development (RAD) of complex systems and applications. With Caché 5, InterSystems delivers a powerful, high-performance, massively scalable database system that supports a plethora of standard protocols, programming languages, and middleware technologies.



Ready, Set, Go!

For our evaluation, we used the Windows and Linux versions of Caché. In both cases, we used the single-user license that can be freely downloaded from InterSystems' Web site. The installation process took only a few minutes and not even the Windows system had to be restarted. All other tools and technologies are installed along with the database system, and the Web server (Internet Information Server or Apache) is reconfigured to incorporate Caché's Web gateway.

After the installation, we started the Caché Studio (see Figure 1), the development environment used for editing the classes stored in Caché, and immediately began to create some classes. Using the integrated wizards, you can easily create classes with all the features available in Caché. Programming Caché's internal language, ObjectScript, is very easy, especially if you are used to coding in Java or JavaScript. If Visual Basic is your favorite programming language, you can also use Caché Basic, a close relative to Visual Basic.

What's Behind the Scenes?

The heart of Caché is its multidimensional data engine with a robust storage system that provides outstanding transactional performance and is capable of serving very large data volumes. Equipped with transactional bitmap indices, queries are evaluated in fractions of a second without sacrificing update and insert performance.

In addition, the multidimensional data model of Caché avoids time-consuming mapping operations between objects and relations because data is natively stored in Caché's multidimensional storage. The Unified Data Architecture (UDA) provides access to the stored data and delivers the necessary metadata information to the relational or object-oriented client. In general, there's no difference if the data is accessed in a relational manner or through objects.

Caché and Java

Our primary focus was on testing the Java features provided by Caché. With three different ways of connecting to Java, Caché provides optimal support for Java developers and programmers. Data stored within Caché can be accessed with SQL via JDBC, and Caché classes can be projected either as Java classes or as Enterprise JavaBeans. Java developers directly gain data persistence without a lot of tedious coding.

The JDBC driver is installed along with the database and is fully compliant with the JDBC standard. To access a Caché class in an object-oriented manner, all we needed to do was specify that it should be projected to Java. This "projection" then automatically generated Java binding classes that could easily be incorporated into Java applications. The great advantage of projections is that the binding classes are automatically generated every time a class is modified and recompiled. So the programmer is relieved of the cumbersome and error-prone process of keeping the database schema and the application bindings up-to-date.

For J2EE application servers, we only had to specify an EJB projection and Caché created entity beans with either bean-managed persistence (BMP) or container-managed persistence (CMP). In contrast to the simple Java projection that creates a single Java class, the EJB projection generates all necessary classes for an EJB entity bean, a deployment descriptor, a deployment command file, and even a little servlet for testing the created bean. We tested Caché's EJB support with a standard installation of JBoss 3.0 and did not experience any difficulties. In addition to JBoss, Caché also directly supports the WebLogic and the Pramati application servers and automatically creates all necessary configuration files for the specific server. Other J2EE-compliant application servers should also work with Caché without any problems.

Summary

We were impressed by Caché's performance, features, and ease of use. We were both very satisfied with our test results and plan to take an even closer look at Caché and its technologies. Note that a more detailed review of Caché, including several code examples, may be found on our Web site (www.synerva.com). As with any software product, there are some trade-offs. In this case, the main one is the learning curve that developers who are only familiar with relational database technology must overcome in order to take full advantage of Caché. InterSystems has done everything they can to ease the transition by providing a development environment that has a familiar look-and-feel and surprisingly complete and usable documentation, but the curve still exists. Also, there are application areas such as data warehousing where market gorillas with traditional relational technology will continue to dominate for the foreseeable future. Caché is worth a very close look.

Mathias Kühn is a Java developer and software architect at SYNERVA, an IT service company for software development, consulting, and training with a special focus on the realization of complex, distributed information systems (www.synerva.com).

kuehn@synerva.com

Christian Och is an IT consultant at SYNERVA and has over 10 years of experience as a software architect and project leader in the areas of heterogenous database integration, distributed object technologies, and application development.

och@synerva.com

Snapshot

Target Audience: Java programmers, database developers

Level: All levels

Pros:

- Combines the object-oriented and the relational paradigm, eliminates the need for an OO-relational mapping
- Easy installation and maintenance, no high-skilled administrator necessary
- Outstanding performance with a small memory footprint
- Support for different Java technologies to access the database (JDBC, Caché object binding, EJB)
- Support for a plethora of technologies and standards
- Embraces common Java standards (Streams, Collections, JDBC, etc.)

Cons:

- No support for polymorphic function signatures
- EJB-, Object-, and JDBC-binding in one JAR file

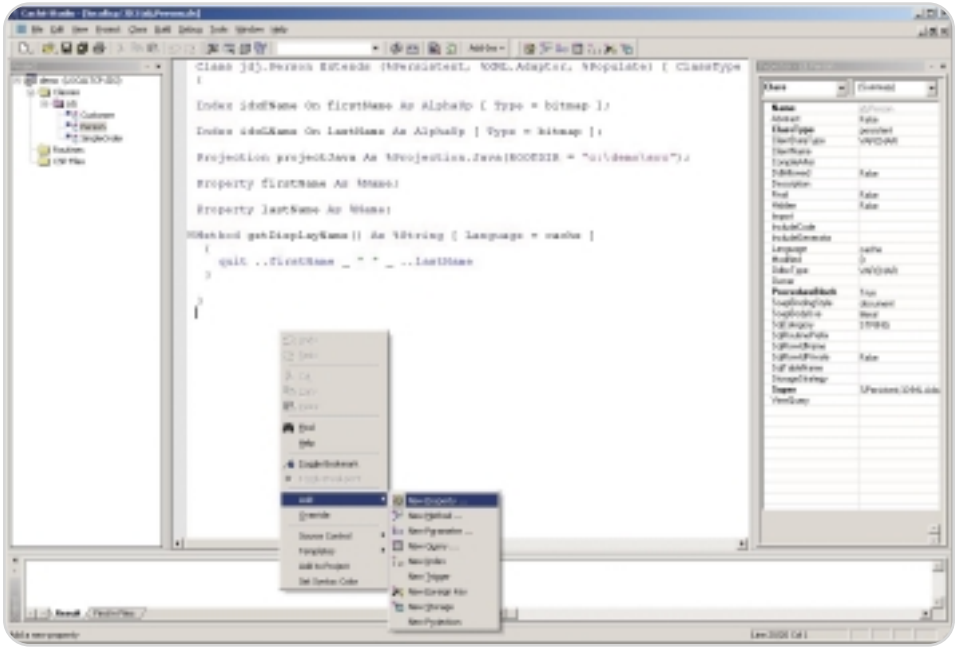


Figure 1 The Caché Studio

From Within the Java Community Process Program

A more open and transparent process

Onno Kluyt

Welcome to the August edition of the JCP column! This month I'll cover a few J2ME-related JSRs and two new JSRs in the J2SE/J2EE environment, but first out of the block is the program office's own JSR, number 215.

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

onno@jcp.org

Further Evolving Rules of the Community

JSR 215 aims to deliver several changes to the rules of the JCP to make the process more open and transparent, both to members and to the general public. These changes should make it easier to determine how a JSR is doing and what the expert group is working on. The plan is to make all draft reviews publicly visible, including the Community Review. The JSR may also introduce the spec lead's choice to have its JSR assigned to both ECs instead of the ME EC or the SE/EE EC. And finally, JSR 215 outlines default requirements for TCKs at the Final Approval Ballot. If all goes well from the time of my writing, JSR 215 should be in Community Review and you should be able to view and provide feedback on the draft process document at http://jcp.org/jsr/detail/215.jsp. The Executive Committee members and I are very interested in hearing your opinions on these proposals.

The J2ME Environment

In the J2ME environment there are a couple of items of note to share with you. JSR 185 successfully passed its Final Approval Ballot. The goal of the JTWI specification is to improve the compatibility, interoperability, and completeness of J2ME technology implementations in mobile phones. The JSR minimizes API fragmentation and raises the bar of functionality for high volume devices in this market.

Nokia has submitted a new JSR that aims to deliver an optional API for scalable 2D vector graphics. Two examples of target use for the API are map visualization and scalable icons. JSR 226 plans to support the W3C Scalable Vector Graphics (SVG) format and is aimed at an MIDP 2.0 environment.

Since the time of writing of last month's column, several J2ME-related JSRs faced reconsideration ballots (JSRs 177, 216, 217, 218, 219). Each JSR successfully passed the ballot, meaning they are approved to progress to the next step in the process. If you're interested in the comments submitted by EC members while voting, visit a JSR's page on JCP.org and click on the ballot link in the status table.

On to the J2SE and J2EE Environments

The last few weeks saw the submission of two new JSRs, one by Oracle and one jointly by Oracle and IBM.

JSR 225, XQuery API for Java, will enable a Java application to submit XQuery queries to an XML data source and process the results of such queries. XQuery is an effort by the W3C XML Query Language Group. While there are APIs such as JDBC, JDO, and RowSets to process relational data, there is no common API to query XML data.

JSR 227, submitted by Oracle, is titled "A standard data access and data binding facility for J2EE". This proposal introduces declarative bindings, a framework of classes that formalize the typical interactions between UI components and the values and methods available on Business Services so that any Java UI rendering technology can declaratively bind to these services. A Business Service is a Java class that publishes and manipulates the objects that represent an enterprise application's persistent data sources.

JavaOne Conference Postscript

The JCP program was quite active at this year's conference. At the "Java Communities in Action" event where you could compare and contrast JINI, JXTA, and the JCP, the program office handed out the first JCP awards. The EC members voted Nokia the Most Innovative J2ME JSR for JSR 184, BEA the Most Innovative J2SE/J2EE JSR for JSR 207, and Jim Van Peursem of Motorola the Best Spec Lead. Jim is the spec lead for JSR 118, an expert group of 60+ companies with more than 120 persons participating, and the JSR still delivered!

During the conference I did the usual number of press interviews. In closing, I'll share with you my favorite question from one of these interviews: "Now that Java has Web services APIs, why do you still have RMI?"

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

International Web Services Conference & Expo

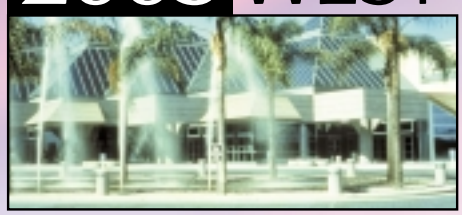
Web Services Edge

3rd Annual

web services conference EDGE &expo

Delivering .NET, JAVA, Mac OS X, & XML Technologies

2003 WEST



SEPT. 30 - OCT. 2, 2003

Santa Clara, CA

web services conference EDGE &expo

REGISTER TODAY!

CALL 201-802-3058

www.sys-con.com

Register before August 1st and

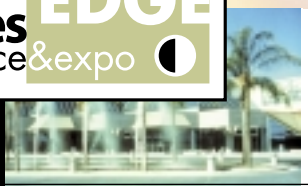
SAVE Up To $300

Produced and Presented: SYS-CON EVENTS

Media Sponsors: WebServices Journal, Java Developer's Journal, WebSphere Developer's Journal, XML Journal, .NET Developer's Journal, WebLogic Developer's Journal, LinuxWorld Magazine, wireless Business Technology, Linux Business Week, C# Advisor, ColdFusion Developer's Journal, PowerBuilder Developer's Journal, marketWIRE, HSPstreet.com, OASIS, SD Times, LinuxWorld.com, SAMS, Open Industry

WEB SERVICES EDGE CONFERENCE & EXPO

web services conference EDGE &expo



SEPT. 30 - OCT. 2, 2003

Santa Clara Convention Center

It is our pleasure to bring the latest edition of the highly successful Web Services Edge Conference & Expo to the Santa Clara Convention Center, September 30 – October 2, 2003. The Third Annual Web Services Edge West Conference & Expo will continue to build on our past success to make available the most current and relevant information to you, our valued attendee.

With the widespread adoption of Web services across the industry, developers are facing new challenges. In this year's conference program, we will address these challenges with our most comprehensive program to date. Web Services Edge 2003 West will provide practical approaches and solutions to overcome the hurdles in developing and deploying Web services in today's competitive markets. Once again Web Services Edge will feature dedicated tracks covering Java, Web Services, .NET, and XML - along with the newly added Mac OS X Track. Sessions in this track will highlight the use of the Mac OS X platform, which combines the ease of use of a Mac with the power of Unix, in applications and Web services development, deployment, and management.

Your three days will include highly informative keynotes, conference sessions, tutorials, industry-leading university certification programs, case studies, and demo presentations. The Expo Hall will be open on September 30 and October 1, featuring the largest grouping of quality exhibitors prepared to field your questions and solve your development needs.

All the best,
SYS-CON Events

FEATURES & ATTRACTIONS

TO ALL CONFERENCE & EXPO REGISTRANTS

- 3 Days Packed with Education and Training
- Keynotes & Panel Discussions from Industry Leaders
- 60 Hard-hitting and Informative Seminars
- FREE .NET Tutorial with Russ' Tool Shed
- Java University Certification Training
- Industry-Leading Certification Programs
- "Birds of a Feather" Discussions
- Round Table Discussions
- Opening Day Welcome Reception
- SAMS Meet the Authors Hot Topics Lounge
- Compelling Case Studies & Best Practices
- Hands-On Labs
- Featured Product Demonstrations
- Exhibit Floor featuring more than 40 companies and hundreds of products
- Real-time SYS-CON Radio Interviews

WHO SHOULD ATTEND

- Software Developer
- Software Engineer
- Development Manager
- Application Developer
- Technical Director
- Analyst/Programmer
- IT Manager
- Technical Architect
- Team Leader
- Software Consultant

HIGHLIGHTED SPEAKERS

Allan Vermeulen

amazon.com

CTO, Amazon.com

CTO and vice president at Amazon.com directly oversees the Platform Technologies group. This group is responsible for guiding Amazon.com's technology architecture, including building and acquiring foundational components. Prior to his move to Amazon.com, Vermeulen was CTO and vice president of development at Rogue Wave Software. He holds a PhD in Systems Design Engineering from the University of Waterloo.

John Schmidt

BEST BUY

Leader of Systems Integration and Middleware, Best Buy Co.

John Schmidt is the chairman of the Methodology Committee for the EAI Industry Consortium and leader of systems integration and middleware at Best Buy Co., a leading specialty retailer of consumer electronics, personal computers, entertainment software, and appliances.

Dave Chappell

sonic SOFTWARE

VP, Chief Technology Evangelist, Sonic Software

Dave Chappell is the vice president and chief technology evangelist for Sonic Software. He has more than 18 years of industry experience building software tools and infrastructure for application developers, spanning all aspects of R&D, sales, marketing, and support services. Dave has also been published in numerous technical journals, and is currently writing a series of contributed articles for *Java Developer's Journal*.

Anne Thomas Manes

burton GROUP

Research Director, Burton Group

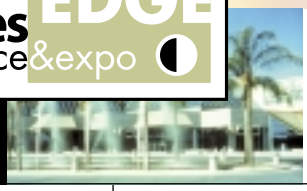
Anne Thomas Manes is a research director at Burton Group, a research, consulting, and advisory firm. Anne leads research for the Application Platform Strategies service. Named one of NetworkWorld's "50 Most Powerful People in Networking" in 2002, and one of Enterprise Systems Journal's "Power 100 IT Leaders" in 2001, Anne is a renowned technologist in the Web services space. Anne participates in standards development at W3C and OASIS. She is a frequent speaker at trade shows and author of numerous articles and the book *Web Services: A Manager's Guide*.

Register Online at

www.sys-con.com

web services conference&expo EDGE

WEB SERVICES EDGE CONFERENCE & EXPO



SEPT. 30 - OCT. 2, 2003

Santa Clara Convention Center

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 Documents: Uses and Differences
- User Interfaces for Web Services
- Web Services Best Practices
- Service Oriented Architecture

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.

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

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

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, and 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 with 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

SPECIAL DISCOUNTS AVAILABLE

Take advantage of the Early Bird and Pre-registration values available right now, or save even more with a group of 5 or more. For special group discounts contact Michael Lynch at mike@sys-con.com, or by phone at (201) 802-3058.

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

PRODUCED BY SYS-CON EVENTS

Conference at-a-Glance

		JAVA	.NET	WEB SERVICES	XML
TUESDAY, SEPTEMBER 30 DAY 1	8:00AM – 4:00PM	**REGISTRATION**			
	9:00AM – 9:50AM	Enterprise Java 1.4	Using WSE 2.0	Web Services Management	Introduction to Xforms
	10:00AM – 10:50AM	Opening Keynote - Allen Vermeulen, CTO, Amazon.com			
	11:00AM – 6:00PM	**EXPO OPEN**			
	2:00PM – 2:50PM	Keynote Panel Discussion - Enterprise Application Integration			
	3:00PM – 3:50PM	Ant Applied in "Real World" Web Services	Smart Devices in Health Care Settings	Service Oriented Architecture	Securing Your XML and Web Services Infrastructure
	4:00PM – 4:50PM	Developing Application Frameworks with SWT	Using the Mobile Internet Toolkit	Web Services Orchestration	XQuery Fundamentals: Key Ingredient to Enterprise Information Integration
	5:00PM	**OPENING NIGHT RECEPTION**			
WEDNESDAY, OCTOBER 1 DAY 2	8:00AM – 4:00PM	**REGISTRATION**			
	9:00AM – 9:50AM	Integrating Java and .NET	Introduction to ROTOR	Security (WS-Security, SAML)	Standards-Based Enterprise Middleware Using XML/Web Services
	10:00AM – 10:50AM	Morning Keynote			
	11:00AM – 4:00PM	**EXPO OPEN**			
	2:00PM – 2:50PM	Keynote Panel Discussion - Interoperability: Is Web Services Delivering?			
	3:00PM – 3:50PM	JUnit: Testing Your Java with JUnit	Using Portable .NET	WS-BPEL	XML and Enterprise Architecture: Technology Trends
	4:00PM – 4:50PM	JDK1.5: The Tiger	ASP.NET with Mono	UDDI: Dead or Alive?	Using XML Schemas Effectively in WSDL Design
	5:00PM – 6:00PM	Squeezing Java	Using WSE with IBM's WSTK	Web Services Choreography, Management, and Security - Can They Dance Together?	Canonical Documents for Your Business: Design Strategies
THURSDAY, OCTOBER 2 DAY 3	8:00AM – 4:00PM	**REGISTRATION**			
	9:00AM – 9:50AM	Using IBM's Emerging Technologies Toolkit (ETTK)	Distributed .NET for Financial Applications	eAI & Web Services	XML and the Fortune 500
	10:00AM – 10:50AM	Morning Technical Keynote			
	11:00AM – 11:50AM	Apache Axis	Developing C# with Eclipse	RPC vs Documents: Uses and Differences	XPath/XSLT 2.0: What's New?
	12:00PM	**BREAK**			
	1:00PM – 1:50PM	Meeting the Challenges of J2ME Development	Windows SharePoint Services	The Seven Habits of Highly Effective Enterprise Service Buses (ESBs)	ebXML & Web Services
	2:00PM – 2:50PM	Keynote Panel Discussion - Summit on Web Services Standards			
	3:00PM – 3:50PM	Empowering Java and RSS for Blogging	BizTalk 2003	See www.sys-con.com for more information	See www.sys-con.com for more information
	4:00PM – 5:00PM	See www.sys-con.com for more information	See www.sys-con.com for more information	See www.sys-con.com for more information	See www.sys-con.com for more information

MAC OS X

	Introducing OS X (Panther) What's New?
	Programming Rich User Interfaces Using Cocoa
	Quick Applications using AppleScript
	Java and OS X: A Perfect Marriage
	Enterprise Java and OS X
	Developing Web Services Using WebObjects
	Cocoa, Carbon, Java: Application Frameworks for OS X (When to use what)
	Securing OS X Applications
	Xserve: Ease of OS X and Power of Unix
	OS X for the Unix Developer
	Introducing Quartz: 2D Graphics for Apple
	See www.sys-con.com for more information

PROGRAM SUBJECT TO CHANGE

Microsoft .net

FREE Microsoft® Tutorial

RUSS' TOOLSHED

TINKERING WITH VISUAL STUDIO

RUSS' TOOL SHED

Join Russ as he shows you how to use Visual Studio .NET

INTRO TO WEB SERVICES USING VS.NET

One of the key ideas behind the .NET strategy is the concept of software as a service, or in short, Web services. This session will explain what a Web service is and provide an overview of its related technologies like XML, SOAP, and UDDI. We will demonstrate how the .NET Framework makes it easy to implement them for new and existing applications. This session will also provide concrete best practices for building XML Web services using Visual Studio .NET. We'll answer many common questions like: How will my Web service scale? How can my XML Web services enable interoperability with Web services from other vendors as well as within my own organization? We'll delve into building highly reliable and secure Web services. Also, we will discuss issues such as dealing with complex data types using WSDL (Web Services Description Language), as well as securing SOAP messages using encryption. We'll see how developers can use enterprise-level XML Web services to simplify customer solutions.

ADVANCED WEB SERVICES USING ASP.NET

This session will explore some of the more advanced areas of SOAP in ASP.NET's support for Web services. ASP.NET Web services are the preferred way for Web developers to expose Web services on the Internet. The goal is quick, easy, and high-performing SOAP services. We will look at how to use the SOAP extension classes to create some very interesting applications on top of the core SOAP architecture found within the .NET Framework. For instance, you can implement an encryption algorithm or screen scraping on top of the Web service call. We'll dig into more advanced topics, explore the SOAP headers, and see ways to ensure security in our Web services.

.NET REMOTING ESSENTIALS

Microsoft .NET Remoting is the .NET technology that allows you to easily and quickly build distributed applications. All of the application components can be on one computer or they can be on multiple computers around the world. .NET Remoting allows client applications to use objects in other processes on the same computer or on any other computer to which it can connect over its network. During this presentation we will discuss what you will need to know to get started with .NET Remoting. We will talk about how .NET Remoting compares with DCOM, how to host remoted objects in a variety of applications, how to call remoted objects from a client application, how to control the lifetime of remoted objects, and how to secure remoting applications.

REGISTRATION FORM

Web Services Edge WEST 2003

SEPT. 30 - OCT. 2, 2003

Santa Clara Convention Center

web services conference&expo EDGE

WEB SERVICES EDGE CONFERENCE & EXPO

CONFERENCE: Sept. 30 – Oct. 2, 2003 **EXPO: Sept. 30 – Oct. 1, 2003**

Santa Clara Convention Center • Santa Clara, CA

THREE WAYS TO REGISTER FOR CONFERENCE

1) On the Web: Credit Cards or "Bill Me." Please make checks payable to SYS-CON Events.
2) By Fax: Credit Cards or "Bill Me" 201-782-9651
3) By Mail: 135 Chestnut Ridge Road, Montvale, New Jersey 07645, Attention: Registration

Please note: Registrations are not confirmed until payment is received.

Please complete sections 1, 2, 3 and 4

1 YOUR INFORMATION (Please Print) ☐ Mr. ☐ Ms.

First Name _____ Last Name _____

Title _____

Company _____

Street _____

Mail Stop _____

City _____

State _____ Zip _____ Country _____

Phone _____

Fax _____ E-Mail _____

2 PAYMENT METHOD: (Payment in full due with registration)

☐ Check or Money Order Enclosed (Registration confirmed upon receipt of payment)

Check #_____ Amount of Check $ _____

Charge my ☐ Visa ☐ MasterCard ☐ American Express ☐ Discover

Name on card_____

Card #_____ Exp. Date_____

Signature_____

Billing Address (if different from mailing address)

_____

3 PLEASE INDICATE YOUR CONFERENCE CHOICE

Total Registration fee $_____

	Before 8/1/03	Before 9/26/03	Onsite
❑ **GP Gold Passport** Good for all three days of the .NET, Web Services, XML, Java, and Mac OS X Tracks, including Keynotes, Panel Discussions, preferred seating for Microsoft .NET's Russ'"Tool Shed" Tutorial, and your choice of one Sun Microsystems Java™ University Class	$1,195.00	$1,395.00	$1,495.00
❑ **3D Three Day Conference** (Does not include Sun Java™ Education)	$1,095.00	$1,295.00	$1,395.00
❑ **2D Two Day Conference** (Does not include Sun Java™ Education) (select any two days: ❑ Tue. ❑ Wed. ❑ Thurs.)	$995.00	$1,195.00	$1,295.00
❑ **1D One Day Conference** (Does not include Sun Java™ Education) (select any one day: ❑ Tue. ❑ Wed. ❑ Thurs.)	$495.00	$595.00	$695.00
❑ **JU1 Sun Java™ University Class** Select one: ❑ Web Services Programming Using Java™ Technology and XML (Sept. 30) ❑ Java™ Fast Path: Programmer (Oct. 1) ❑ Java™ Fast Path: Architect (Oct. 2)	$595.00	$695.00	$795.00
❑ **JU2 Sun Java™ University Class** Select two: ❑ Web Services Programming Using Java™ Technology and XML (Sept. 30) ❑ Java™ Fast Path: Programmer (Oct.1) ❑ Java™ Fast Path: Architect (Oct. 2)	$1,095.00	$1,295.00	$1,395.00
❑ **JU3 Sun Java™ University Class** Select three: ❑ Web Services Programming Using Java™ Technology and XML (Sept. 30) ❑ Java™ Fast Path: Programmer (Oct. 1) ❑ Java™ Fast Path: Architect (Oct. 2)	$1,195.00	$1,395.00	$1,495.00
❑ **EO Expo Only**	FREE	FREE	$50.00

4

A. Your Job Title

❑ CTO, CIO, VP, Chief Architect
❑ Software Development Director/Manager/Evangelist
❑ IT Director/Manager
❑ Project Manager/Project Leader/Group Leader
❑ Software Architect/Systems Analyst
❑ Application Programmer/Evangelist
❑ Database Administrator/Programmer
❑ Software Developer/Systems Integrator/Consultant
❑ Web Programmer
❑ CEO/COO/President/Chairman/Owner/Partner
❑ VP/Director/Manager Marketing, Sales
❑ VP/Director/Manager of Product Development
❑ General Division Manager/Department Manager
❑ Other (please specify)_____

B. Business/Industry

❑ Computer Software
❑ Computer Hardware and Electronics
❑ Computer Networking & Telecommunications
❑ Internet/Web/E-commerce
❑ Consulting & Systems Integrator
❑ Financial Services
❑ Manufacturing
❑ Wholesale/Retail/Distribution
❑ Transportation
❑ Travel/Hospitality
❑ Government/Military/Aerospace
❑ Health Care/Medical
❑ Insurance/Legal
❑ Education
❑ Utilities
❑ Architecture/Construction/Real Estate
❑ Agriculture
❑ Nonprofit/Religious
❑ Other (please specify)_____

C. Total Number of Employees at Your Location and Entire Organization (check all that apply):

	Location	Company
10,000 or more	01 ❑	01 ❑
5,000 - 9,999	02 ❑	02 ❑
1,000 - 4,999	03 ❑	03 ❑
500 - 999	04 ❑	04 ❑
100-499	05 ❑	05 ❑
100 or less	06 ❑	06 ❑

D. Please indicate the value of communications and computer products and services that you recommend, buy, specify, or approve over the course of one year:

❑ $10 million or more
❑ $1 million - $9.9 million
❑ $500,000 - $999,999
❑ $100,000 - $499,999
❑ $10,000 - $99,999
❑ Less than $10,000
❑ Don't know

E. What is your company's gross annual revenue?

❑ $10 billion or more
❑ $1 billion - $9.9 billion
❑ $100 million - $999 million
❑ $10 million - $99.9 million
❑ $1 million - $9.9 million
❑ Less than $1 million
❑ Don't know

F. Do you recommend, specify, evaluate, approve or purchase wireless products or services for your organization?
01 ❑ Yes 02 ❑ No

G. Which of the following products, services, and/or technologies do you currently approve, specify or recommend the purchase of?

❑ Application Servers
❑ Web Servers
❑ Server Side Hardware
❑ Client Side Hardware
❑ Wireless Device Hardware
❑ Databases
❑ Java IDEs
❑ Class Libraries
❑ Software Testing Tools
❑ Web Testing Tools
❑ Modeling Tools
❑ Team Development Tools
❑ Installation Tools
❑ Frameworks
❑ Database Access Tools / JDBC Devices
❑ Application Integration Tools
❑ Enterprise Development Tool Suites
❑ Messaging Tools
❑ Reporting Tools
❑ Debugging Tools
❑ Virtual Machines
❑ Wireless Development Tools
❑ XML Tools
❑ Web Services Development Toolkits
❑ Professional Training Services
❑ Other [Please Specify]

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3058 by September 16, 2003.

CANCELLATIONS, SUBSTITUTIONS, REFUNDS

Fax written request to SYS-CON Registration 201-782-9651. Requests for refunds received prior to August 29, 2003 will be honored, less a 10% handling charge; requests received after August 29, 2003, and before September 12, 2003, will be honored less a 20% handling charge. No requests for refunds will be honored after September 12, 2003. Requests for substitutions must be made in writing prior to September 26, 2003. No one under 18 is permitted to attend. No warranties are made regarding the content of sessions or materials.

Speakers, sessions, and schedule are subject to change without prior notice.

No solicitation by anyone other than official exhibitors, sponsors or marketing partners is permitted. Such behavior is cause for expulsion without refund.

JAVA DEVELOPER'S JOURNAL Advertiser Index

Advertiser	URL	Phone	Page
Asperon Corporation	www.asperon.com		25
Borland Software Corporation	go.borland.com/j6		11
Canoo Engineering AG	www.canoo.com/ulc/	41 61 228 94 44	7
Colorado Software Summit	www.softwaresummit.com		16
ESRI	www.esri.com/mapobjectsjava	888-332-2320	9
Fair, Isaac & Company	www.fairisaac.com/rules		Cover III
InetSoft Technology Corp.	www.inetsoft.com/jdj	888-216-2353	17
Infotectonica	www.infotectonica.com		34
Infragistics, Inc.	www.infragistics.com	800-231-8588	14-15
InstallShield Software Corp.	www.installshield.com/puzzlejdj		21
iSavix	http://isavix.net	703-689-3190	41
LinuxWorld Magazine	www.sys-con.com	888-303-5282	43
Northwoods Software Corp.	www.nwoods.com/go	800-434-9820	58
Oracle	oracle.com/switch	800-633-0538	4
OracleWorld	oracle.com/start	800-355-5808	35
Parasoft Corporation	www.parasoft.com/jdj08	888-305-0041	19
Quest Software, Inc.	http://java.quest.com/jclass/jdj		13
Quest Software, Inc.	http://java.quest.com/performasure/jdj		33
Quest Software, Inc.	http://java.quest.com/jprobe/jdj		Cover IV
RealObjects	www.realobjects.com		23
RefactorIT	www.refactorit.com		45
ReportingEngines	www.reportingengines.com/info/ere3.jsp	888-884-8665	24
Software FX	www.softwarefx.com		27
Sonic Software	www.sonicsoftware.com		Cover II
SpiritSoft	www.spiritsoft.com/climber		31
WebAppCabaret	www.webappcabaret.com/jdj.jsp		29
Wily Technology	www.wilytech.com	1-888-GET WILY	39
Web Services Edge West 2003	www.sys-con.com	201-802-3069	59-64
Zero G	www.zerog.com	415-512-7771	3

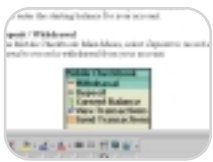
Next Month

Extending JAAS

User authentication and access control are important security measures for most Java applications, especially J2EE applications. The Java Authentication and Authorization Service (JAAS), the core API of J2SE 1.4 and 1.5, represents the new security standard. This article will introduce the JAAS framework, identify the key components and their roles in the authentication and authorization process, and explain how the authentication and authorization perform.

Java and Stream Ciphers

In the 1990s, I worked with the Winsock 2 interface and encryption when they first came out from Microsoft in Beta form; that was exciting as they allowed you to easily encrypt data through the networks. When Java Sockets came out, I was hooked because the encryption could be easily managed through a stream of data.



This article discusses streams from the cipher perspective and provides an example of how to design and build a stream algorithm so you can practice proper techniques rather than relying on the technology to do it for you.

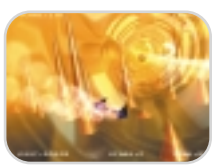
Java Games Development

In Part 2 of this three-part series, the discussion focuses on the 85% portability goal for Java, features of the Java platform that are a better choice for games development than other platforms, and which features are missing from the API that are essential (or would just make life easier) for games development.

Aligo Omni Mobile Platform

Way back in ***JDJ*** (Vol. 6, issue 11) I took a look at Aligo's M-1 Mobile Application Server. Back then, the technology market was still in the heady "bubble" days and wireless was the next big frontier. Well, fast-forward to mid-2003 and here we are wallowing in the throes of an extended technology slump. However, wireless applications are still a part of the next big frontier, and the team at Aligo has been hard at work updating their software. I recently got the chance to get my hands dirty with the latest, forthcoming release of the Aligo Omni Mobile Platform.

Cashing In: Distributing and Marketing Your J2ME Killer App



You finally put the finishing touches on your new J2ME application and are probably starting to think about what is involved in marketing and distributing your killer app. It is at this point that, all too often, great ideas are left as nothing more than great ideas. From concept to consumer, this article will explain all that is involved in getting your application certified and distributed by the world's largest J2ME distributors.

Twenty-First Century Snake Oil Salesmen

Software is created by programmers who write code, testers who try and break the code before users do, and analysts who are incapable of either task. Analysts know this and like a congressman's PR agent on their lunch break, they must constantly adapt to find new ways to remain on the payroll. The answer in IT is no different than any similar dilemma in which a person finds himself: bluff, fraud, and deceit.

For inspiration, the analyst thinks back to those days at school when he or she sat in science classes and gazed at complicated diagrams being drawn by the teacher. The kids who understood what the teacher was saying went on to become proper engineers, while the befuddled analysts instead picked up the subliminal message "complicated diagrams = good". The syllogism was clear – if they too could create presentations that others couldn't understand perhaps one day they could assume the same authoritative role over the audience that the science teacher enjoyed.

Acting on this childhood experience, the analyst buys books about analysis patterns and sets about to confuse and overwhelm all around him. Expensive tools are bought that don't create any code or contribute to the finished software product, but nevertheless print out reams of paper and create meaningless diagrams. Other analysts and managers nervously guffaw compliments like the emperor's subjects in the Hans Christian Andersen fable. Data modeling becomes trendy, methodologists and process reengineers are hired, while authors of expensive and pathetic hardback books dazzle gullible conference attendees who are mesmerized by presentation foils like first year music students at a John Cage concert.

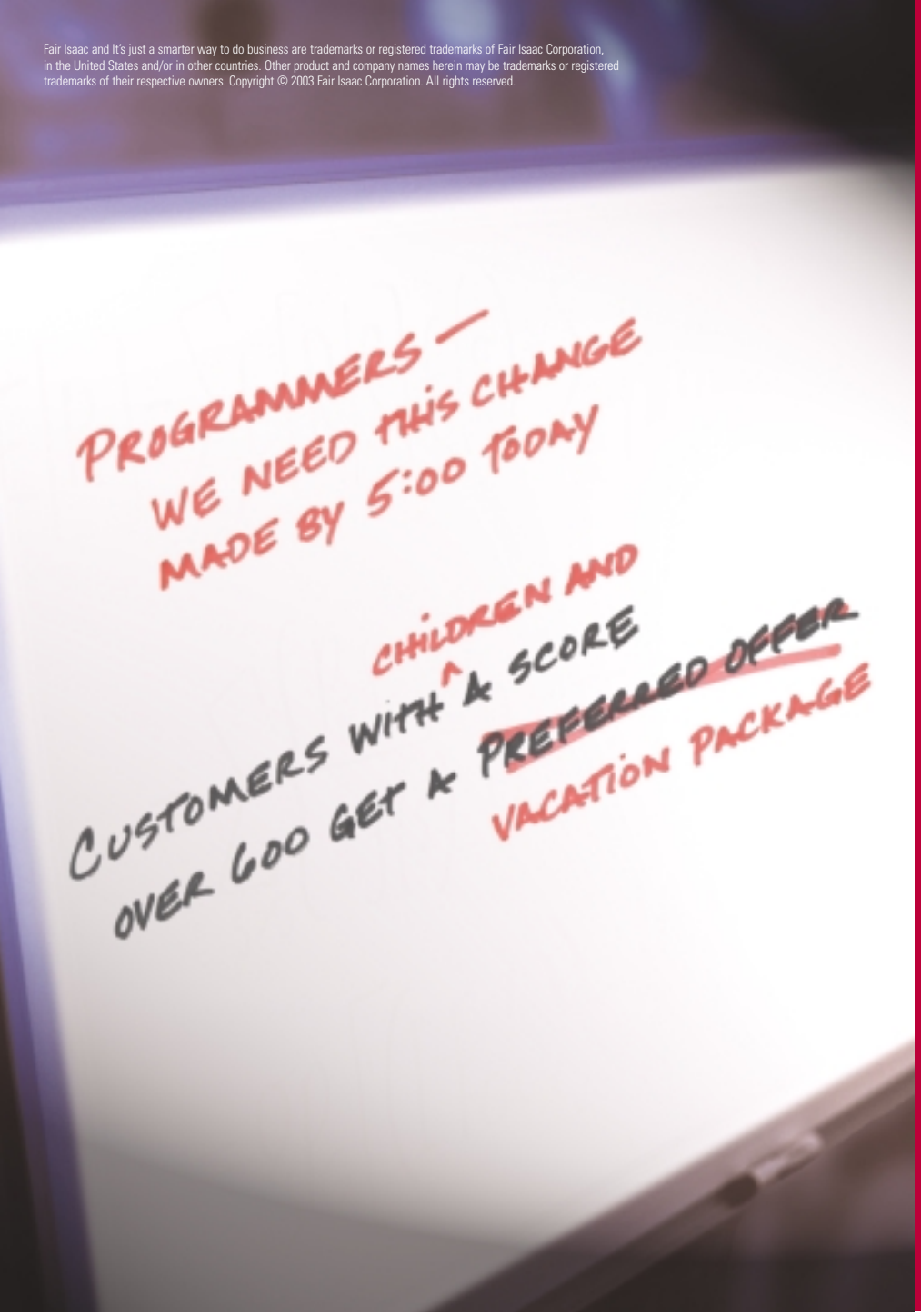
None of this, however, helps users get their software any sooner. It does have a noticeable advantage over coding, because analysis occurs at the start of a project where time and money are plentiful and the enthusiasm of senior management and users still exists. Just when coding is about to begin, the whole methodology landscape may change, and the same person who one day drew multidimensional Shlaer-Mellor data universes, overnight becomes an expert at the new techniques and wastes valuable project dollars and time convincing senior management that arguments about OO notation and design patterns are now the key to project success. A professional software analyst is a master of trend moulting, chasing bandwagons faster than a Republican party candidate.

Fortunately, it finally looked like analysts were going to be run out of town when grass root programmers gained popular acceptance for Extreme Programming (XP). This put the power back into the hands of the coders and testers who now deal directly with users, create the simplest code possible to get the job done, and do it lots of times.

XP gave me great hope that the overthrown analysts would become the Troy McClures of software development, with perhaps a few survivors becoming electric tongue scraper salesmen on late night infomercials. With this in mind, I recently attended a presentation by a famous analyst "who shall not be named" who began by saying that in the past 10 years, misguided analogies between civil engineering and software had created the mess we had all lived through. Waiting for him to admit he played more than his fair share in the whole charade, I instead had to redefine my definition of irony when he then espoused his own particular variant of XP, tried to sell his recently published book, and attempted to convince us that you had to employ his consulting skills to make sure the methodology was being followed correctly.

I often wonder whether the FBI department investigating bogus claims by companies selling penis-enlargement pills should perhaps turn their attention to professional software analysts. Same scam, different scenario.

Java Dudes



CAN YOUR CODE KEEP UP WITH YOUR BUSINESS GROUP?

Fair Isaac Blaze Advisor lets you create, deploy and modify the rules that drive your business—quickly, easily and cost-effectively. Maintained separately from the rest of your system code, our business rule technology gives you unprecedented visibility into and control over logical decisions used throughout your enterprise. Save time, save money, save yourself headaches. Visit our Web site for more information about Fair Isaac Blaze Advisor business rules management. ***It's just a smarter way to do business.™***

FairIsaac®

www.fairisaac.com/rules

Introducing JProbe® 5.0
…now smarter and faster than ever

JProbe®

Find the cause of J2EE code performance, memory and threading problems faster than ever before with JProbe 5.0. New investigative features for finding memory problems combined with dramatic performance improvements mean even quicker resolution of problems in your application, servlet, JSP and EJB code.

JProbe Suite

JProbe Profiler

JProbe Memory Debugger

JProbe Threadalyzer

JProbe Coverage

QUEST SOFTWARE®

For more info and a free eval, visit:

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