

JAVA™ DEVELOPER'S JOURNAL

The World's Leading Java Resource

April 2003 Volume:8 Issue:4

JAVADEVELOPERSJOURNAL.COM

Show Report



Hynes Convention Center, Boston, MA page 68

From the Editor

Alan Williamson pg. 5

Guest Editorial

Nigel Thomas pg. 6

J2EE Editorial

Ajit Sagar pg. 16

J2SE Editorial

Jason Bell pg. 36

J2ME Editorial

Glen Cordrey pg. 50

Show Report

pg. 68

Java News

pg. 72

RETAILERS PLEASE DISPLAY UNTIL JULY 31, 2003

\$9.99US \$9.99CAN



SYS-CON MEDIA



SWT

A NATIVE WIDGET TOOLKIT FOR JAVA

PAGE 58

WRITTEN BY STEVE NORTHOVER AND JOE WINCHESTER

Swing GUI Components: Abbot - A Friendly JUnit Extension for GUI Testing *An effective framework* **Satadip Dutta** **8**

Feature: Debunking the Myth of In-Process App Layer Caching in J2EE Architectures *Alternative solutions* **Helen Thomas** **18**

Open Source: Taking Continuous Integration to the Next Level *Continuous what?* **Franz Garsombke** **30**

Feature: J2ME: A European Point of View *A wireless wonderland for Java* **Sven Haiges** **52**

JDJ Labs: Tifosi 2002 Enterprise Integrator *by Fiorano Software* **Laurent Michalkovic** **58**

JDJ Labs: Conquer-IT! J2EE *by Trans-World Resources, LLC* **Tim Rondeau & Duncan Murphy** **62**

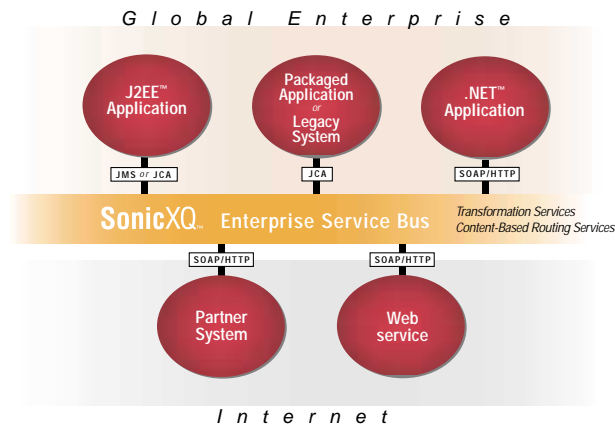
JSR Watch: Inside the JCP *Eliminating the confusion regarding the process* **Alan Williamson** **66**

From the Inside: The McNealy Windscale of Performance Benchmarks *A new battle cry* **74**

Everybody's focused on exposing applications as Web services while letting someone else figure out how to connect them. We're that someone else.

Introducing **SonicXQ**TM

Integrate with ease, extend at will.TM



Most companies are focused on exposing individual applications as Web services, but haven't yet considered how to effectively integrate or manage them. And that's no small task.

SonicXQ is the first product to deliver on the vision of the Enterprise Service Bus: a standards-based, service-oriented architecture that reliably and securely integrates enterprise applications through Web services and JCA technologies. SonicXQ includes configurable XML processing

services that allow applications and business partners to easily interact over the SonicXQ bus, eliminating the need for point-to-point integrations and the expensive centralized solutions that implement them.

And because SonicXQ is built upon the proven SonicMQ[®] messaging backbone, it provides end-to-end reliability and comprehensive security between services. Our patent-pending Dynamic Routing ArchitectureTM ensures that

Web services and distributed processes can scale to meet the demands of today's global enterprises.

Capitalize on the promise of Web services right now with SonicXQ. After all, what good is a Web service if it's an island unto itself?



sonic
SOFTWARETM
www.SonicSoftware.com/JDJ



Why do industry leaders choose Zero G?

Developers already know that Zero G's InstallAnywhere® and PowerUpdate® are critical for their software installation and updating needs. But, buying from us means you get more than just our award-winning multi-platform software deployment solutions. You also get our customer service, our world-class technical support, and best of all, our people - a team of skilled professionals who are committed to the partnership that starts the minute you download one of our products. That's why industry leaders like Sun Microsystems, Novell and Borland choose us.

ZERO G

Your software deployment partner

www.ZeroG.com

INTERNATIONAL ADVISORY BOARD

CALVIN ALUSTIN (Lead Software Engineer, J2SE Linux Project, Sun Microsystems)
JAMES DUNCAN DAVIDSON (JavaServlet API/MIP API, Sun Microsystems)
JASON HUNTER (Senior Technologist, CollabNet)
JON S. STEVENS (Apache Software Foundation)
BILL ROTH (Group Product Manager, Sun Microsystems)
BILL WILLETT (CEO, Programmer's Paradise)
BLAIR WYMAN (Chief Software Architect IBM Rochester)

EDITORIAL

EDITOR-IN-CHIEF: ALAN WILLIAMSON
EDITORIAL DIRECTOR: JEREMY GEELAN
EXECUTIVE EDITOR: NANCY VALENTINE
J2EE EDITOR: AJIT SAGAR
J2ME EDITOR: JASON R. BRIGGS
J2SE EDITOR: JASON BELL
PRODUCT REVIEW EDITOR: JIM MILBERY
FOUNDING EDITOR: SEAN RHODY

PRODUCTION

PRODUCTION CONSULTANT: JIM MORGAN
ASSOCIATE ART DIRECTOR: LOUIS F. CUFFARI
ASSOCIATE EDITORS: JAMIE MATUSOW
GAIL SCHULTZ
JEAN CASSIDY
JENNIFER STILLEY
ONLINE EDITOR: LIN GOETZ
TECHNICAL EDITOR: BAHADIR KARUV, PH.D.

WRITERS IN THIS ISSUE

JASON BELL, GLEN CORDREY, SATADIP DUTTA, FRANZ GARSOMBKE,
SVEN HAIGES, LAURENT MICHALKOVIC, DUNCAN MURPHY,
STEVE NORTHOVER, TIMOTHY RONDEAU, AJIT SAGAR, HELEN THOMAS,
NIGEL THOMAS, ALAN WILLIAMSON, JOE WINCHESTER, BLAIR WYMAN

SUBSCRIPTIONS

FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,
PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT

SUBSCRIBE @ SYS-CON.COM

COVER PRICE: \$5.99/ISSUE
DOMESTIC: \$69.99/YR. (12 ISSUES)
CANADA/MEXICO: \$99.99/YR. OVERSEAS: \$99.99/YR.
(U.S. BANKS OR MONEY ORDERS)
BACK ISSUES: \$10/EA., INTERNATIONAL \$15/EA.

EDITORIAL OFFICES

SYS-CON MEDIA, 135 CHESTNUT RIDGE RD., MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9600

JAVA DEVELOPER'S JOURNAL (ISSN#1087-6944) is published monthly
(12 times a year) for \$69.99 by SYS-CON Publications, Inc.,
135 Chestnut Ridge Road, Montvale, NJ 07645.

Periodicals postage rates are paid at Montvale, NJ 07645
and additional mailing offices. POSTMASTER: Send address changes to:
JAVA DEVELOPER'S JOURNAL, SYS-CON Publications, Inc.,
135 Chestnut Ridge Road, Montvale, NJ 07645.

© COPYRIGHT

Copyright © 2003 by SYS-CON Publications, Inc. All rights reserved.
No part of this publication may be reproduced or transmitted in any
form or by any means, electronic or mechanical, including photocopy or
any information storage and retrieval system, without written permission.
For promotional reprints, contact reprint coordinator Carrie Gebert,
carrieg@sys-con.com. SYS-CON Media and SYS-CON Publications, Inc.,
reserve the right to revise, republish and authorize its readers
to use the articles submitted for publication.

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



ALAN WILLIAMSON EDITOR-IN-CHIEF

A Day in the Sun

I'm sitting here in the San Francisco airport waiting for a flight home after spending a few days out here with Sun. I met with representatives from the complete Java spectrum including the main man, Mr. Gosling. It was a good and very worthwhile trip and the one thing I can safely report is that Sun is back! They were a little lost over the past few years, with the poor developer feeling left out in the cold. But they are making major inroads to win back the developer and focus on what they do best – technology.

James Gosling commented that it just became too overwhelming for their engineers to cope with the volume, with James himself fighting some 10,000 e-mails that were vying for attention in his inbox. So they retracted into their shell and answered nobody! The only people you could talk to were the marketing and PR people. No matter who you were. But I can safely report that this is no longer the situation; Sun is open for business again.

I spoke with the main spec lead for J2EE, Mark Hapner, who took me through the upcoming 1.4 release of J2EE, which encompasses Web services support. Mark spoke candidly about the need for WSI integration to J2EE and it more than justified delaying the edition. Many J2EE vendors have been implementing their own WS offering, locking the J2EE developer into a particular application server. The time had come to bring together all the experience and knowledge and incorporate it into the main platform. Now you'll be able to have your EJB exposed as a Web service as easily as a servlet, but developed in such a way that it's completely J2EE compliant.

I then spoke to the Desktop Java team, who were responsible for making sure Swing was a serious alternative for the development of desktop client tools. They've been spending a lot of time on the performance of

Swing and now feel they have a serious offering. They sat me down in front of a number of demo apps, including one with a complete XP look-and-feel. I was impressed, until I realized they were showing it on a 2GHz machine. Anything is fast on that sort of machine! Fortunately, they had brought another laptop with them that was a little more realistic and it was acceptable. They did comment that they are presently looking at the time it takes for Java to start up on the desk and trying to improve that. Ironically, it wasn't the startup speed I was looking at, but the actual feel and snappiness of the GUI controls – all very reactive.

One of the new things that will be introduced soon is the ability for Java to update itself with the latest version, something similar to the Windows updater. This is to ensure that as many people as possible are always on the latest version, thus giving Java the best possible chance.

Next it was J2ME. I was taken through the new standards to come out of MIDP 2.0 and where they were driving that particular movement. With the number of Java handsets now outnumbering desktop PCs, there was a buzz of excitement. That said, it's important not to get too excited with that number, as it's a bit like Apple saying they now have the largest Unix user base in the world; the majority of their users are completely oblivious to the fact that they are indeed part of that statistic.

I met with the JCP folks and you can find how that went within this issue of **DDJ**. On the whole, I had a lot of information thrown at me and I still have to sit down and listen through all the meetings again, so expect more information coming at you very soon. I would like to thank Laura Ramsey and Corina Ulescu, who between the two of them ensured my day went smoothly and that I met everyone I needed to. Thank you.

Catch me on my blog. ☎

AUTHOR BIO

When not answering your e-mails and working on the next issue of DDJ, Alan heads up a small team dubbed the "Thunderbirds of the Java industry" providing on- and offsite 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



Is J2EE Too Big for Its Own Good?

WRITTEN BY NIGEL THOMAS

Speaking to Sun's J2EE marketing team recently, we learned that J2EE 1.4 has been delayed so that "vital" new Web services features could be added. Originally targeted for the second half of 2002, J2EE 1.4 FCS is now not expected until this summer.

J2EE is perhaps the most significant of the three Java platform "editions" – Micro, Standard, and Enterprise. It's usually J2EE that is stacked up against .NET in the marketplace. Delays to J2EE releases significantly impact on the extent to which Enterprise Java can maintain and improve its market penetration.

I question whether Sun's current monolithic approach to the Enterprise Edition is either appropriate or effective. On the one hand, J2EE 1.4 is just another set of specifications going through the Java Community Process (JCP); like all standards processes, that's bound to mean compromises and delays as competitors sit around the table to hammer out the details. But J2EE is a big beast, as we've all learned from bitter experience:

- Big projects tend to move at the speed of the slowest task.
- Coping with problems caused by unstable and changing dependencies is always best avoided if possible.

Although it may suit the giants of the industry – BEA, IBM, Oracle, and Sun – to have a megaspecification, this discriminates against specialist vendors who cannot benefit from the compliance test suite for an individual component, but are legally bound to license the entire J2EE stack – all or nothing. That could be seen as an excessively restrictive barrier to entry; in these hard times, small vendors simply can't afford to meet the same licensing fees as the big players. Some simply evade formal licensing and compliance, while others are driven out of J2EE development entirely. Sun is now offering open-source projects free-of-charge access to the compliance tests, which will further squeeze out the smaller vendors who historically have driven innovation in this industry.

How Can We Redress the Balance?

Developers increasingly base products and solutions around just those subsections

of the J2EE platform that they need, and they don't like being forced to pay for components, such as EJB, that they don't use.

Forward-thinking infrastructure developers are building plug-and-play Java platforms that make it possible to assemble your own style of server. Servlet or EJB container doesn't quite suit your needs? Then assemble your own purpose-built container!

Look at how the JBoss microkernel is blossoming; take a glance at Jakarta's Avalon and Phoenix projects. This is the way of the future – modular, easily customizable, and extensible server frameworks, on top of which architects can combine subsets of J2EE with components like Java Data Objects (JSR-12), the Java Rule Engine API (JSR-94), or JCache (JSR-107) that are – for the moment at least – outside the J2EE boundary.

Mix and Match

With the Mobile Edition (J2ME), sheer physical constraints forced the acceptance of different horses for different courses; J2ME's configurations and profiles ensure that for every requirement the right platform can be assembled, without undue fragmentation of the standard.

I propose that for J2EE 1.5, Sun and the JCP should isolate the minimum kernel required to support enterprise infrastructure. Perhaps JCP can learn from Extreme Programming practices and organize a series of easier-to-control "sprints" rather than another 18–24 month marathon. The "optional extras" like EJB, JSP/servlets, JDBC, and JMS can each have an independent-release cycle; each one must be transparently pluggable into any vendor's core J2EE kernel; no vendor specifics allowed. At the same time, Sun should change its J2EE licensing policy to a "per-component" basis, priced accordingly.

Architects can then easily assemble the configurations they need – such as the frequently touted J2WE or Web Edition – reducing the costs of development but retaining all the advantages of interchangeable standards-based components. Each component can be upgraded much more quickly, and "Write Once, Run Anywhere" comes another step closer. ☛

nigel.thomas@spiritsoft.com

Author Bio

Nigel Thomas is director of product management at SpiritSoft.

SYS-CON MEDIA

PRESIDENT AND CEO
FUAT A. KIRCAALI fuat@sys-con.com
VICE PRESIDENT, BUSINESS DEVELOPMENT
GRISHA DAVIDA grisha@sys-con.com

ADVERTISING

SENIOR VICE PRESIDENT, SALES AND MARKETING
CARMEN GONZALEZ carmen@sys-con.com
VICE PRESIDENT, SALES AND MARKETING
MILES SILVERMAN miles@sys-con.com
ADVERTISING SALES DIRECTOR
ROBYN FORMA roybri@sys-con.com
ADVERTISING ACCOUNT MANAGER
MEGAN RING megan@sys-con.com
ASSOCIATE SALES MANAGERS
CARRIE GEBERT carrieg@sys-con.com
KRISTIN KUHNLE kristen@sys-con.com
ALISA CATALANO alisa@sys-con.com

EDITORIAL

EXECUTIVE EDITOR
NANCY VALENTINE nancy@sys-con.com
ASSOCIATE EDITORS
JAMIE MATUSOW jamie@sys-con.com
GAIL SCHULTZ gail@sys-con.com
JEAN CASSIDY jean@sys-con.com
JENNIFER STILLEY jennifer@sys-con.com
ONLINE EDITOR
LIN GOETZ lin@sys-con.com

PRODUCTION

PRODUCTION CONSULTANT
JIM MORGAN jim@sys-con.com
LEAD DESIGNER
LOUIS F. CUFFARI louis@sys-con.com
ART DIRECTOR
ALEX BOTERO alex@sys-con.com
ASSOCIATE ART DIRECTOR
RICHARD SILVERBERG richards@sys-con.com
ASSISTANT ART DIRECTOR
TAMI BEATTY tami@sys-con.com

WEB SERVICES

VICE PRESIDENT, INFORMATION SYSTEMS
ROBERT DIAMOND robert@sys-con.com
WEB DESIGNERS
STEPHEN KILMURRAY stephen@sys-con.com
CHRISTOPHER CROCE chris@sys-con.com

ACCOUNTING

ACCOUNTS RECEIVABLE
KERRI VON ACHEN kerri@sys-con.com
FINANCIAL ANALYST
JOAN LAROSE joan@sys-con.com
ACCOUNTS PAYABLE
BETTY WHITE betty@sys-con.com

SYS-CON EVENTS

PRESIDENT, SYS-CON EVENTS
GRISHA DAVIDA grisha@sys-con.com
CONFERENCE MANAGER
MICHAEL LYNCH michael@sys-con.com

CUSTOMER RELATIONS

SENIOR CUSTOMER CARE/CIRCULATION SPECIALIST
NIKI PANAGOPOULOS niki@sys-con.com
JDJ STORE MANAGER
RACHEL MCGOURAN rachel@sys-con.com



Developing Web Services before WebLogic Workshop.



Developing Web Services after WebLogic Workshop.

If getting dinner via a spear strikes you as less than efficient, you'll appreciate BEA WebLogic Workshop™. This unique development framework eliminates tedious and time-consuming steps, while you build Web services, create new applications and tackle the toughest integration challenges. Plus, you'll have all the support of our dev2dev developer community. To see how BEA WebLogic Workshop can radically simplify your Web services development, download it today at dev2dev.bea.com/useworkshop.



The fastest route to enterprise Web services.

Abbot — A Friendly JUnit Extension for GUI Testing



WRITTEN BY
SATADIP DUTTA

Graphical user interface (GUI) testing is a potentially problematic area because constructing effective test cases is more difficult than the corresponding application logic. The roadblocks to effective functional GUI testing are:

- Traditional test coverage criteria like “80% coverage of the lines of code” may not be sufficient to trap all the user interaction scenarios.
- End users often use a different user task interaction model than the one conceived by the development team.

Functional GUI testing needs to deal with GUI events as well as the effects of the underlying application logic that results in changes to the data and presentation.

The common methods for functional GUI testing are the “record and execute” script technique and writing test programs for different scenarios. In the “record and execute,” the test designer interacts with the GUI and all the events are recorded in a script. The script can later be replayed to re-create user interactions for a particular scenario. In the test programs, the test designer tries to understand and write tests for the various GUI decision points.

This article discusses how Abbot can be used to quickly and effectively come up with a comprehensive GUI test framework for Swing GUI applications. Abbot (<http://abbot.sourceforge.net/>) is a JUnit extension for Swing GUI testing. It provides an interesting framework that can be used for test case generation as well as “record and execute” scripting.

Introduction to Abbot

Abbot builds upon the `java.awt.Robot` class to provide an automated event generation and validation framework for Swing GUI components. The framework can be used to create, record, and execute scripts and programmatic test cases in Java. Abbot also has a script editor called Costello that facilitates the creation of scripts in XML. The frame-

work can also be well integrated to run with JUnit.

To illustrate the use of Abbot to create a GUI test infrastructure, this article will make use of the following two scenarios.

1. **GUI already exists:** The GUI has been coded and the application needs to be tested, but no unit tests are available. This scenario will primarily make use of the “record and execute” style and then focus on how test suites can be created and run with JUnit.
2. **GUI has not been coded:** The GUI has been designed on paper; however, no working code exists. This scenario will primarily focus on creating test cases in Java. This method will create a homogeneous suite of tests both for the back-end application code and the GUI.

The two scenarios focus on the end-point cases in functional GUI testing. The principles mentioned can be mixed and matched to suit the needs of the project. To get started with Abbot, download the JAR files and place them in the `lib` directory of the project.

Scenario One: GUI Already Exists

In this scenario, most of the GUI has already been coded. Automated testing is not available and the testing is mainly being done by hand. The goal is to come up with a test suite in the shortest time possible so that the quality metrics for the project can be met effectively.

The script editor Costello will be used to develop a battery of tests. Costello provides the “record and execute” functionality that will allow us to record different user interaction scenarios with the GUI and test those scenarios efficiently. The scripts that record the user interac-

An effective framework

tions are saved as XML. The salient features of the scripts that help in rapidly creating a functional test framework are:

- **Focuses on test creation:** The XML-based scripts can be written rapidly (directly or by using Costello) and are dynamically interpreted by Abbot. There’s no need to write new code and compile test cases.
- **Focuses on validating GUI decision points:** Abbot allows the test creators to add assertions to check for values resulting from user interactions. The `assert` tag is used to check for valid results from a user interaction before the script proceeds. This feature is very useful, as it helps pinpoint the step in which the GUI fails and aids in regression testing.
- **Provides infrastructure to create robust scripts:** Some GUI “record and execute” tools produce very fragile scripts. These tools not only store the events but also the GUI component position information, such as coordinates, to identify them. These scripts are very fragile because any change in positional attributes (changes in layout or running the script on a different platform) breaks the scripts. Abbot uses a number of attributes to dynamically identify the component without depending on any positional attributes. These attributes in aggregate form a component reference, which can be used within the script even before the component itself necessarily exists.
- **Provides a high-level abstraction over the Java Robot class:** Abbot builds on top of the `Java Robot` class by providing a layer of abstraction. The abstraction layer allows end users to create test cases that can capture high-level semantics like the OK button, Select



Does your software give you a better way to monitor and manage application development? Ours does.

Our software goes to great lengths to help you optimize application development projects, so you won't have to. AllFusion™ is Computer Associates' comprehensive and flexible application lifecycle management solution. Whether your projects are in .NET, JAVA or the mainframe, you can transparently monitor and manage all phases of application development, including people, process, and tools. And that means you won't just finish projects on time. You'll deliver them within budget. ca.com/allfusion

AllFusion™  Computer Associates®

CO from the list of states, etc. This makes the scripts easy to understand and enhances them in the future.

An example of a test script in XML is provided in Listing 1.

With this background about the features of the script, it's time to look into how to use the script editor to create the test cases. The sample GUI shown in Figure 1 is an example from the Java Swing tutorial available at java.sun.com.

The GUI is a Celsius-to-Fahrenheit converter that takes in a numeric value and on the button click displays the converted result in Fahrenheit. To test this GUI, we can write some simple GUI tests to verify the results on the input of a positive number, a negative number, and a nonnumeric input.

The script editor will be used to create the test cases. Costello, the script editor, can be started by:

```
java -jar lib/abbot.jar
```

Once the GUI is running, the first step is to create a new script (File->New Script). Each script has a launch line that specifies the method (containing the GUI) to be tested. The launch line is edited with the information that can launch the GUI. The launch information

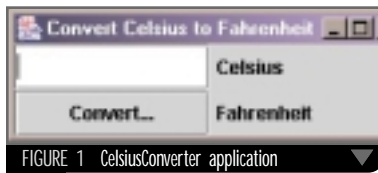


FIGURE 1 CelsiusConverter application

requires the method name, arguments for the method, class information, and the classpath (see Figure 2).

```
method="main"
args=""
class="CelsiusConverter"
classpath="src/demo"
```

Note that the XML test script example has the same information in the launch tag.

With the launch information in place, the GUI can be launched (Test->Launch). This will prepare the framework to record user actions, press F2 (Capture->All Actions). With this step we can record the first case, which inputs a positive number, and then press convert.

Once the user interactions have been captured, we can move the actions out of the sequence block for better readability. With the script looking the way we want, it's now time to add an assertion. To do this, we'll use the Hierarchy tab to navigate to the GUI component. Upon selection of the GUI component, the name-value pair to be validated can be selected and the Assert Property = Value button will add the assertion (see Figure 3).

The other two test cases can be created similarly to form a test suite. The test cases can run within the JUnit test harness utilizing either the command line or the GUI test runner. The junit.extensions.abbot.ScriptFixture class is subclassed to create the CelsiusConverterTest.

```
public class CelsiusConverterTest
    extends ScriptFixture
```

The junit.extensions.abbot.ScriptTestSuite class is used to autogenerate a suite based on test scripts matching a certain criteria. In this case, all scripts residing in a particular directory will be used. The test ScriptTestSuite class is subclassed from the junit.Framework.TestSuite class.

```
public static Test suite() {
    return new ScriptTestSuite
        (CelsiusConverterTest.class,
        "src\demo\scripts\CelsiusConverter");
}
```

The main method of the CelsiusConverterTest class invokes the JUnit test runners.

```
public static void main(String[]
args){
    args = Log.init(args);
```

```
String[] names = {
    CelsiusConverterTest.class.getName()
};
if (args.length == 1 &&
    args[0].equals("--gui"))
    junit.swingui.TestRunner.main(names);
else
    junit.textui.TestRunner.main(names);
}
```

This approach facilitates the creation of a variety of user interaction scenarios that can be easily integrated with JUnit. Another interesting and handy feature is the ability to insert existing scripts into another script. Some high-level GUI tasks can be broken down into smaller tasks. In these cases, I've created scripts for the most granular tasks and then created different interaction scenarios by inserting the scripts for different tasks in a different order or by adding additional user interactions in between the tasks. Since the scripts are XML-based, they're very easy to read and they help in rapidly generating different scenarios.

Scenario Two: GUI Is Ready to Be Coded

In this scenario, the GUI tasks have been identified and the specification is ready to be coded. A good way to begin coding is to write the tests that need to be passed before the GUI is created. The XML-based scripts can be used to create different test cases; however, to demonstrate the full power of Abbot, this section will discuss how the JUnit test cases can be written for the GUI.

The Abbot Java API will be used to write traditional JUnit test cases. The salient features of the API that help in creating test cases are:

- **Standard JUnit test case structure:** Every test case is a subclass of junit.framework.TestCase. The test case has a setUp() and tearDown(), and the tests are structurally similar to JUnit.
- **Provides the ability to simulate user events:** Swing GUI testing is different from testing standard Java classes. For standard Java classes, the JUnit framework can run in a separate thread and invoke the methods on the Java classes. However, JUnit doesn't provide a framework that meets the testing needs of Swing applications. For Swing applications, the tests need to wait for the GUI to run and then execute the different user-generated events. Abbot provides the ComponentTester class. This allows the test case to invoke the GUI and gener-

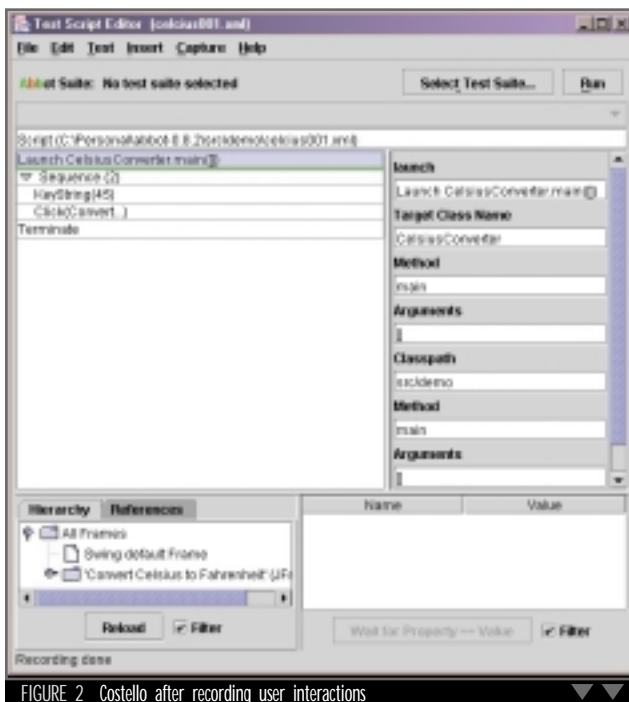


FIGURE 2 Costello after recording user interactions

Thanks to LISA my butt isn't so big.

- Cecilia Dufils, QA Analyst



I used to sit for hours on end trying to write test code. Not anymore. With LISA, I now have more free time to do other things, like workout to Iron Buns 4. Unlike other products on the market, LISA is the only testing tool I can actually use. She codes my test cases for me. She supports unit, regression, and load testing; Windows, OSX, Linux, and Solaris; plus websites, web services, and Enterprise Java. And she only costs as low as \$59 per developer seat. You go girl.

LISA, real automated testing for J2EE. FREE 30 day trial at www.itko.com/lisa

ate user events on the AWT thread of the application (simulating a user interaction), without having to explicitly perform any synchronization. The flow of the test code is a simple series of user actions and GUI state checks.

- **Provides component-specific testers:** Provides classes to simulate the various events that are specific to different Swing components. For example, JMenuItemTester would enable the simulation of various user-generated events within the test for each of the JMenuItem's in the GUI. These classes are available in the abbot.test package.

The Celsius-to-Fahrenheit converter example presented in the previous section is used to walk through the steps involved in creating a test case using the Abbot Java API. In this test case, the input of a negative number to the Celsius input will be tested.

Like JUnit test cases, the CelsiusConverterTest class will subclass the junit.Framework.TestCase class and the constructor will initialize the name of the test.

```
public class CelsiusConverterTest
    extends TestCase {
    public CelsiusConverterTest
        (String name) { super(name); }
}
```

The setUp() and tearDown() methods can be used to initialize the frame in which the GUI will run. The setUp() method may be used to populate any arrays or other defaults that are required

by the GUI to be fully functional.

```
private ComponentTester tester;
protected void setUp() {
    tester =
        ComponentTester.getTester
            (CelsiusConverter.class);
    // wait for the window to be displayed
    tester.waitForFrameShowing
        ("Convert Celsius to
        Fahrenheit")
}
```

To write the test, we create a method called testNegativeNumberInput(). This test method needs to get a reference to each of the GUI components that are being tested. One way Abbot gets a reference to a GUI component is by looking up a component of the appropriate class. This allows Abbot to continue to refer to different GUI components even if the layout changes.

```
ComponentReference ref =
    new ComponentReference
        ("tempCelsius", JTextField.class);
Component tempCelsius =
    getFinder().findComponent(ref);
```

The next step is to initialize the ComponentTester so that the user interaction event of typing in a negative number can be generated

```
tester = ComponentTester.
    getTester(tempCelsius);
tester.actionKeyString(-45);
```

Once the sequence of user interactions is in place, we would need to verify if the result is correct by adding assertions. The reference to the label is obtained and then the value of the label is tested using assertEquals()

```
JLabel fahrenheitLabel =
    (JLabel)getFinder().findComponent(ref)
;
assertEquals("-49 Fahrenheit",
    fahrenheitLabel.getText());
```

To run the test, the TestHelper class available in the test package in Abbot will be handy. The TestHelper class also provides automatic test suite generation functionality.

```
public static void main(String[]
args){
    TestHelper.runTests
        (args,
```

```
CelsiusConverterTest.class);
}
```

This method can be used in conjunction with JUnit test cases for the application logic to create a homogeneous test suite. The decision points in the application logic can be tested using normal JUnit tests. Test cases generated using Abbot will test for different user interaction scenarios.

Conclusion

Writing functional GUI tests can be a tedious task. The concepts from both of the testing scenarios mentioned can be combined to produce a comprehensive suite of tests. Abbot as a framework provides both scripting functionality and a Java API. The framework easily integrates with the JUnit test harness and therefore, during application development, the functional GUI tests can become a part of the test suite. All these features of Abbot make it an effective framework for rapidly creating a comprehensive test framework.

Resources

- Abbot: <http://abbot.sourceforge.net>
- JUnit: www.junit.org
- Abbot Java API: <http://abbot.sourceforge.net/doc/api/index.html>
- Java Swing Tutorial: <http://java.sun.com/docs/books/tutorial/uiswing>

sdutta@vt.edu

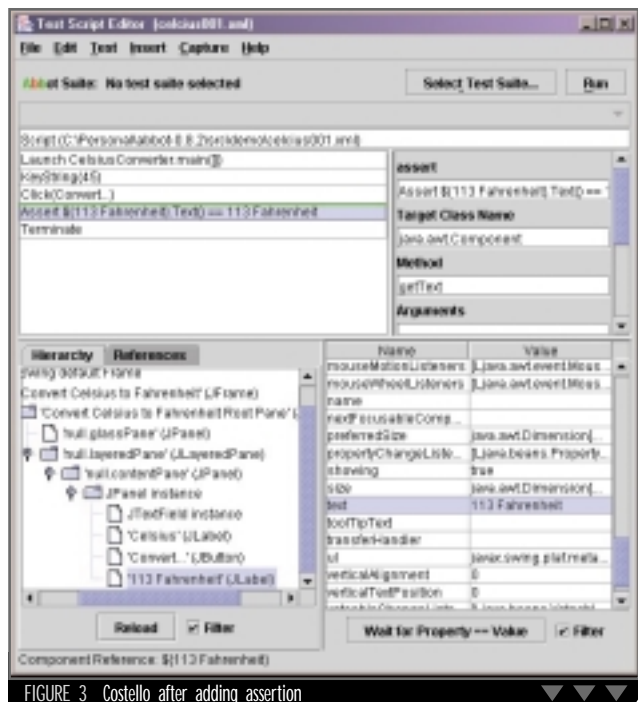


FIGURE 3 Costello after adding assertion

Listing 1: Example of a Test Script

```
<AWTTestScript>
<component id="Convert Celsius to
    Fahrenheit Root Pane"
    class=
        "javax.swing.JRootPane"

    index="0"
    title=
        "Convert Celsius to
        Fahrenheit"
    tag="Convert Celsius to
        Fahrenheit Root Pane"/>
...

<launch classpath="src/demo"
args="[]"
class=
    "CelsiusConverter"
method="main" />
<action args="45"
method=
    "actionKeyString"/>
<action class=
    "javax.swing.
    AbstractButton"
args="Convert..."
method="actionClick"/>
<assert value=
    "113 Fahrenheit"
    component=
        "113 Fahrenheit"
    method="getText"/>
<terminate/>
</AWTTestScript>
```

Download the Code! www.javadevelopersjournal.com

XML Management By Design



QoB™ LAB

Introducing the QoB™ Lab and the QoB™ Assistant.
Pre-emptive XML network management starts with the right design.

Day 1:

XML Heterogeneity

XML is standard, but the implementation of XML is not. Understand the choices that need to be made in regard to Java and .NET, and across major standards.

XML Messaging

Making sense of the XML messaging fundamentals, including: communication models, layers, and requirements. Hands-on session: create feature/benefit analysis and best design for your organization.

XML Data Payloads

Focus Session. Examine the data payloads of XML messages in your industry, address the technical issues, and employ best practices.

XML Interoperability

XML and Web services standards are being created for your industry, and across all industries. How will this impact your architecture? Session ends with a payload choice exercise.

Day 2:

QoB™ Assistant

Tools and strategies for pre-empting interoperability issues and tracking them in Service Oriented Architectures.

XML & Web Services Security

Security has special needs in an XML and Web Services environment. This session covers security fundamentals including: security obstacles, cryptography, authentication, authorization, accountability, and administration, with special consideration of current and proposed standards. The session closes with a use case, and design guidance for security integration in enterprise architectures.

Basic XML Network Management

Understand the basic approaches for XML management, design tips, and strategies for using common management services with today's standards.

Day 3:

Advanced XML Network Management

Review the evolution of XML Service Networks with emphasis on architectural characteristics. Design your management lifecycle stack that covers planning, provisioning, security, and management.

Service Oriented Architecture Readiness

Investigate the challenges of exposing legacy systems using XML, such as re-use in a federated environment. Learn the value of a roadmap for your services oriented architecture.

Web Services Architectures

Review the design and implications of XML and Web Services architectures. Examine the approaches and uses of application servers, integration suites, intermediaries, and enterprise buses. Close with strategies and trade-offs.

Action Plan

Learn the top 10 things you can do to manage XML in your organization—by design.

Special call to insurance. Check out our insurance specific QoB Lab that covers ACORD XML!

Take the QoB Lab Test Drive.

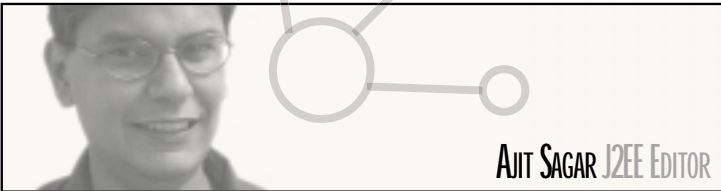
If you are responsible for developing, architecting, or deploying XML and Web Services solutions within your organization, take 5 minutes to see how Swingtide can help you get ahead of the curve.

Please visit www.swingtide.com/testdrive.

Swingtide reserves the right to modify sessions.



SWINGTIDE



AJIT SAGAR J2EE EDITOR

How EE Is Your J2?

My 2 1/2 year old son has a birth certificate on his door that says “native Texan.” Now I’ve lived in Dallas for several more years than those he has covered in his short stint on this planet, but that doesn’t make me a native Texan. I am in a strange state of flux right now. I am originally from India, have lived in Dallas for about 11 years and have a house there, am living out of Denver for a few months, and I spend five working days in Milwaukee every week. While my son is a full-compliant U.S. citizen, I can proclaim some compliance. I guess I can also call myself a Milwaukee resident.

There are varying degrees of compliance, and the definition of compliance in any environment is very much dependent on how much leeway that environment offers you. On page 6 (in this issue of *JDJ*), Nigel Thomas talks about how J2EE is becoming too big as a platform and how this creates unnecessary cost and complexity for assembling specialized applications. One of the steps that Sun had taken toward simplifying the complexity of the Java platform was splitting the platform into three editions – J2EE, J2SE, and J2ME. Now, J2EE is just a layer on top of J2SE and therefore is not really an autonomous edition of the platform.

The J2EE Blueprints provide a definition of what it takes to be a J2EE application. In a nutshell, if your application is comprised of J2EE components, then it is a J2EE application. And your component is a J2EE component if it runs inside a system-level enterprise entity called a J2EE container. Customization of the application is achieved by a deployment descriptor. The container can be a Web container, an EJB container, and so on. However, when you design your application, it’s usually neither feasible nor desirable to convert all existing components to run in J2EE containers. After all, the majority of the

systems were written before Java came along and there’s no way everything will be converted into a Java component.

However, J2EE allows your application to be called a “J2EE application” in several ways. The Blueprints describe several application scenarios ranging from applications that use both the Web and EJB components, through applications that are built on only one of the two types of components, to standalone clients that use neither Web nor EJB components. If your application falls into any of these categories, it’s a J2EE application. Typically, enterprises rarely use all the component types offered by J2EE. So, when an organization decides to move their legacy systems to a J2EE environment, what is it that falls under such initiatives? If they develop standalone clients that talk to legacy systems via some Java wrappers, does that make the application suite a suitable candidate to be called a “J2EE application”? Or if they develop a Web front to mainframe systems, does that constitute a J2EE application? Or if applications running a JVM communicate with legacy systems via a messaging bus (not JMS), does that make it an application worthy of the Java 2 Enterprise Edition? How much of the E in the J2EE does it really take?

From Sun’s perspective, a Java drop in a non-Java ocean is a feather in the cap, and an organization is announced as another satisfied customer that has happily adopted J2EE. However, the benefits of the J2EE platform are really evident when the Java distributed object model and the Web components become a part of the platform migration.

As for Web services, obviously this new paradigm for achieving application integration has made its way to the J2EE umbrella. Hence the delay in the release of the next version of the platform. Is it time for Sun to take another stab at reducing the complexity by splitting up the J2EE platform? Is it time for a J2WSE? ☛

ajit@sys-con.com

AUTHOR BIO

Ajit Sagar is the J2EE editor of *JDJ* and the founding editor of *XML-Journal*. He is the director of engineering at Controlling Factor, a leading B2B software solutions firm based in Dallas, and is well versed in Java, Web, and XML technologies.

How EE Is Your J2?

There are varying degrees of compliance, and the definition of compliance in any environment is very much dependent on how much leeway that environment offers you.

by Ajit Sagar

16

Debunking the Myth of In-Process Application Layer Caching in J2EE

Architectures

J2EE applications are characterized by the continuous creation, consumption, and destruction of various types of application objects. A popular solution that addresses the costs of object creation and destruction is to store these objects in the application process memory, often referred to as in-process caching. This article demonstrates that in-process caching can be harmful to application performance and scale in memory-constrained application environments due to its adverse impact on GC performance.

by Helen Thomas

18

Taking Continuous Integration to the Next Level

Sometimes, the easiest and most rewarding part of development is the actual coding.

Managers and developers often dismiss the building, deploying, testing, and metrics-gathering aspects of the software life cycle.

Continuous integration is the concept of automating the build cycle so that code is built and tested many times during the day. This allows developers to integrate their code into the build daily or hourly, thus lowering the chances of integration problems.

by Franz Garsombke

30

Application Server Experts Agree

“Enterprises that are seeking a well-appointed midtier solution will definitely want to give Oracle9iAS a test drive.”



“Oracle9iAS is now a formidable application server.”



“Oracle: one stop app server shop.”



“[With Oracle9iAS,] Oracle is delivering what enterprises need to build a robust, scalable ebusiness infrastructure.”



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

ORACLE®

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

**oracle.com/experts
or call 1.800.633.1072**

Copyright © 2002, Oracle. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. Reprinted from eWeek, September 2, 2002, with permission. Copyright © 2002 Ziff Davis Media Inc. All rights reserved.

Debunking

the Myth of In-Process Application Layer Caching in

J2EE Architectures

The case for external
object storage

written by Helen Thomas

J2EE applications are characterized by the continuous creation, consumption, and destruction of various types of application objects.

These objects may be product objects in e-commerce applications, session objects, or user profile objects, to name a few common examples. Creation and destruction of these objects is expensive – object creation usually requires accessing persistent storage in back-end systems (e.g., DBMSs and file systems), while object destruction requires releasing resources used by the object (e.g., memory, database connections, etc.).

A very popular solution to address the costs of object creation and destruction is to store these objects in the application process memory, often referred to as in-process caching. One of the greatest advantages of in-process caching is that it provides fast access to application objects, which can improve application performance. Unfortunately, there are drawbacks associated with in-process caching. By far, the greatest drawback is its impact on garbage collection (GC) overhead in the application process where the cache resides. In certain situations, in-process caching can significantly increase the costs of GC, resulting in severely increased CPU utilization and response times.

In this article, I demonstrate that in-process caching can be harmful to application performance and scale in memory-constrained application environments due to its adverse impact on GC performance. I then discuss an alternative caching approach, an external caching architecture, and demonstrate that it can provide significant performance benefits for such applications when compared to in-process caching. Before delving into the details, I first provide an overview of two key concepts that are central to understanding this article: *object caching* and *Java memory management*.

Object Caching Basics

Object caching refers to storing an object that has been generated for a particular request so it can be used to serve subsequent requests for the object. A commonly used type of

object caching is in-process caching, in which objects are stored in the application process memory. By keeping objects in local memory, subsequent requests for the object can be satisfied directly from memory, reducing the overhead of object creation and destruction.

Figure 1 depicts an in-process caching architecture. Nearly all enterprise software systems (whether coded from scratch or running packaged applications) have multiple caches that run inside each application process. Most J2EE application servers offer in-process caching features. In fact, in-process caching has become so widely used that it is considered the de facto standard approach for optimizing application performance and scale. JCACHE (JSR #107), which proposes a standard Java API for in-process caching, is evidence of this.

Overview of Java Memory Management

This article is concerned with two key concepts in Java memory management: *garbage collection* and *reference objects*.

Garbage Collection

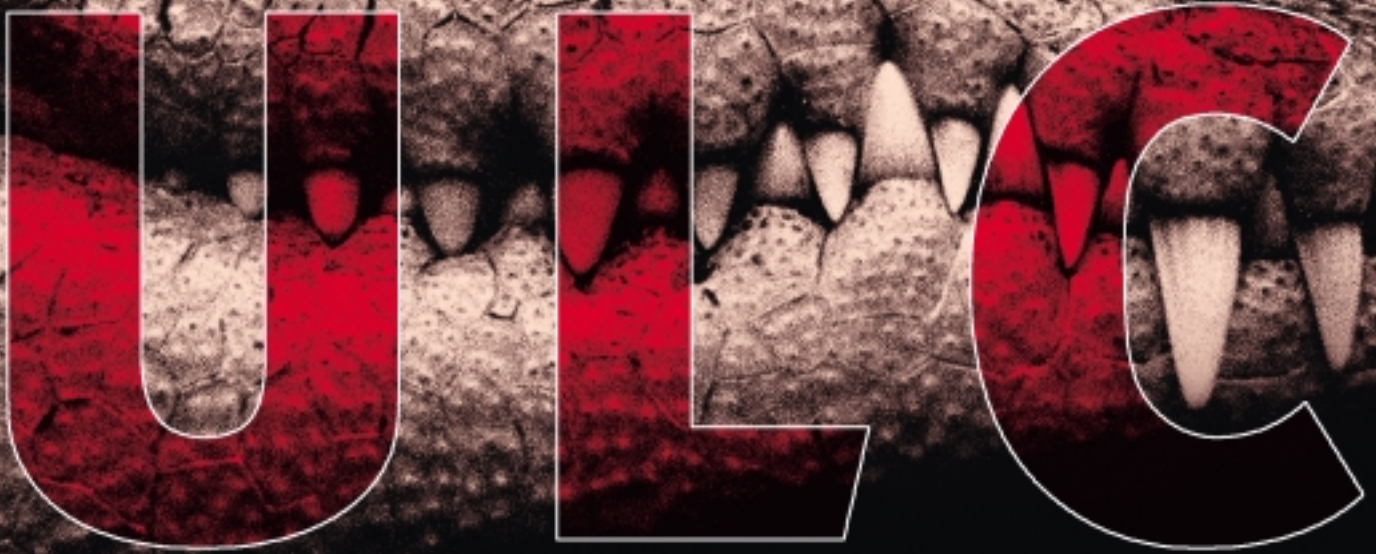
Garbage collection is the method by which memory is automatically reclaimed from unused objects. An object is eligible for GC when it can no longer be reached from any pointer in the running program. Most JVMs use a generational collection model, which takes advantage of the fact that, in most programs, the vast majority of objects are very short-lived (e.g., temporary data structures). In a two-generational collection scheme (as used in the HotSpot and JRockit JVMs), a young generation is maintained for short-lived objects and an old generation for long-lived objects. When the young generation fills up, a minor collection is invoked. When the old generation fills up, a major collection is invoked.

Since major collections require iterating over all living objects, they take orders of magnitude longer to complete than minor collections. This point results in two key conclusions regarding GC performance:

1. The longer an object survives, the more collections it will endure and thus, the slower GC becomes.
2. By arranging for most objects to be collected via minor collections, GC can be very efficient.

NEW!
Release 5.1
out now!

Java thin clients made easy



ULC

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 Tel +41 61 228 94 44
Kirschgartenstrasse 7 Fax +41 61 228 94 49
CH-4051 Basel <http://www.canoo.com/ulc/>
Switzerland ulc-info@canoo.com

Download your free trial today!

Reference Objects

A reference object encapsulates a reference to some other object so that the reference itself may be examined and manipulated like any other object. Two types of reference objects, soft and hard references, are of interest here, since they are often used for in-process caching. A hard reference object is an application object that can be reached directly, i.e., without traversing any reference objects, by a pointer in the running program. A soft reference holds a reference to one or more objects – the objects referred to are called soft reference objects. Figure 2 depicts these two types of reference objects in application memory.

A unique characteristic of soft reference objects is that they can be reclaimed by the garbage collector if additional application memory is needed. For this reason, soft references are typically used to cache objects that are “not critical” to the application. As an example, consider a product object in an e-commerce application. If the product object is not available in memory when requested, the cost to the application is the cost to re-create the object from scratch. Hard references, on the other hand, are usually used to cache “critical” objects, i.e., objects whose loss would render the application unresponsive (or, at best, severely delayed) to requests. An example of such a critical object would be a session object.

Impact of In-Process Caching on GC Performance

A set of tests was conducted to examine the impact of in-process caching on GC performance. Both soft and hard references are examined in these tests.

Test Description

The test application is a Java-based shopping portal that maintains a user profile for each user in order to serve personalized content. Three test cases are considered initially:

1. **No Cache:** This is the baseline case, in which no caching is used. A request is served by instantiating the profile object (requiring a database access), generating the requested page, and then writing the updated profile information to the database.
2. **Soft Reference:** The profile object is kept in memory for the duration of the session, or until the object is evicted from the cache due to the replacement policy, whichever occurs first. The cache is implemented using soft references.
3. **Hard Reference:** Same as the Soft Reference case, except that the cache is implemented using hard references.

The basic test configuration consists of an application server, a profile database, and a

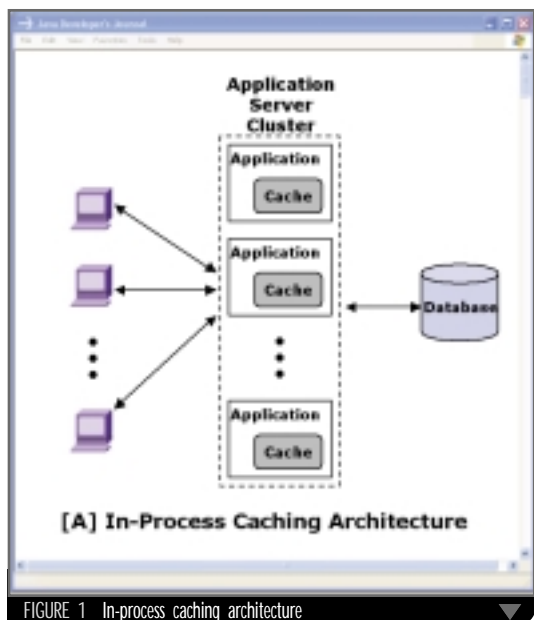


FIGURE 1 In-process caching architecture

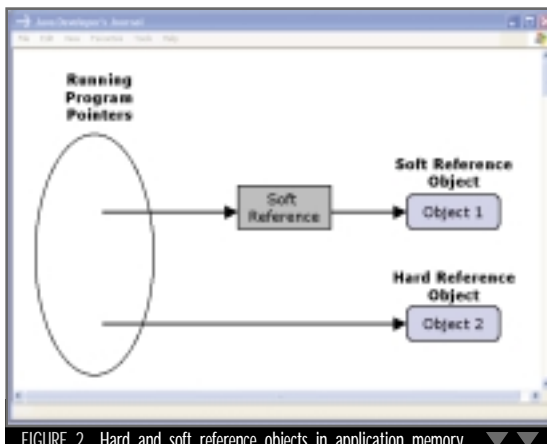


FIGURE 2 Hard and soft reference objects in application memory

Module

Software
Hardware

Application Server

WebLogic 6.1
Solaris 2.6
JDK 1.3.1 (server-side HotSpot enabled)
Sun 420R, quad-processor (450 Mhz), 2 GB RAM

Profile Database

Oracle 8.1.7
Solaris 2.6
Sun 420R, quad-processor (450 Mhz), 2 GB RAM

Load Simulator

LoadRunner
Windows 2000
Dell Optiplex, PIII 700 Mhz processor, 512 MB RAM

TABLE 1: Testing infrastructure

PARAMETER	VALUE
Average size of profile object (KB)	100
JVM memory size (MB)	64
Application footprint (MB)	12.5
Total available memory size (MB) = JVM memory size – application footprint	51.5
Duration of experiment (hours)	1
System load (requests per second)	275

TABLE 2: Testing parameter values

cluster of clients, similar to the architecture shown in Figure 1. The application server is WebLogic, the profile database is Oracle, and the client load simulation software is LoadRunner. Detailed hardware and software specifications are shown in Table 1. All modules reside on the same local area network and communicate via sockets.

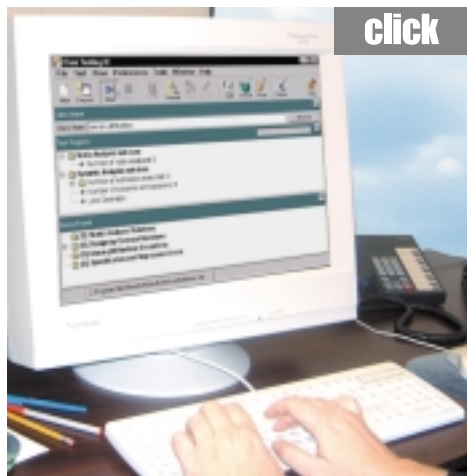
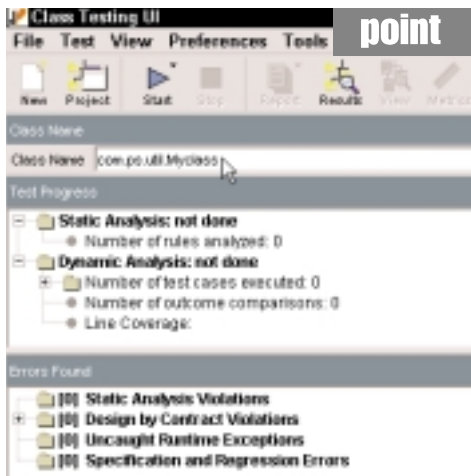
The key test parameters include the average size of a profile object (100KB), JVM memory size (64MB), and total available memory size (51.5MB). A complete list of test parameters is displayed in Table 2. In the two caching cases, the cache is initially empty. Measurements are recorded once the cache is 50% full to ensure that the system is in steady state. The system load of 275 requests per second is a sufficient load to create resource contention for the experiments.

Test Results

Figure 3 shows CPU utilization versus time (the curve labeled External will be discussed subsequently). In the No Cache case (see Figure 3A), the CPU utilization remains fairly constant at about 70%, with several peaks at various points. These peaks represent the points when the GC process runs. The smaller peaks (reaching about 75%) represent minor collections, while the taller peaks (reaching about 90%) represent major collections.

The system behavior is quite different when in-process caching is used. In the Soft Reference case (see Figure 3B), the CPU utilization is relatively

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

Special Offer:

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

Copyright ©2003 Parasoft Corporation. All rights reserved. All Parasoft product names are trademarks or registered trademarks of Parasoft Corporation in the United States and other countries. All other marks are the property of their respective owners.

Platforms:

- Linux
- Solaris
- Windows 2000/XP

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



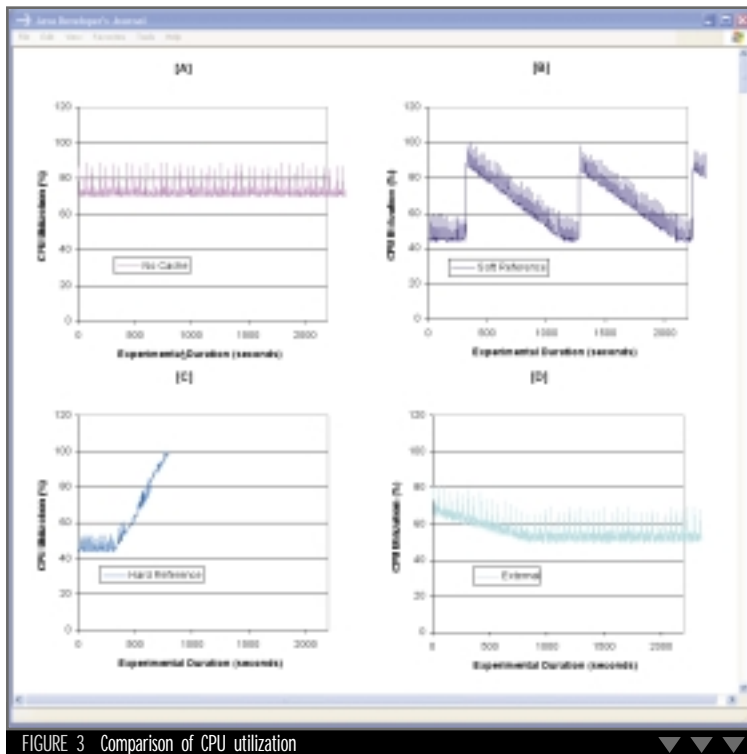


FIGURE 3 Comparison of CPU utilization

low initially, around 45%, with periodic peaks representing minor and major collections. At time 350 seconds, a sudden spike in CPU utilization occurs. This spike indicates the point at which the system reclaims memory from the soft reference objects, since it's unable to reclaim sufficient memory through minor or major collections. This reclamation empties the cache. As a result, each subsequent request requires that the profile object be generated. This work, along with the corresponding cache operations (e.g., insertion, lookup), translates into the dramatic increase in CPU utilization shown. As the cache fills again, CPU utilization decreases.

In the Hard Reference case (see Figure 3C), the behavior is initially similar to the Soft Reference case. However, in the Hard Reference case, the system is unable to collect the needed memory, and thus, a system crash ultimately results.

Figure 4 shows the average response times, which mirror the CPU utilization results. The response time for the No Cache case remains fairly constant at about 145 milliseconds (ms). For the Soft Reference case, the response time is initially at 70ms and decreases to about 10ms as the cache fills. The response time then jumps to about 200ms, indicating the point when the soft reference objects are reclaimed. The Hard Reference case follows a similar pattern, except that the

response time continues to increase until the system crashes.

A comparison of the three cases provides some interesting observations. Table 3 provides a comparison of the in-process caching cases with the No Cache case for selected experimental ranges. The Soft Reference case can provide up to a 36% reduction in CPU utilization and up to a 14x reduction in response times when compared to the No Cache case. However, these improvements are only possible when the system has sufficient memory so that soft reference objects don't need to be reclaimed (e.g., during time 0-349 seconds). When soft reference objects need to be collected, the Soft Reference case actually degrades performance - up to a 43% increase in CPU utilization and a 38% increase in response times (e.g., during time 350-680 seconds).

The use of in-process caching increases the frequency of major collections. With the Soft Reference case, major collections occur about three times as often as in the No Cache case (about once every 30 seconds for the Soft Reference case and about once every 100 seconds for the No Cache case).

The Hard Reference case provides similar improvements as the Soft Reference case as long as sufficient memory is available. However, once memory becomes scarce, the Hard Reference case also degrades performance when compared to the No Cache case - up to a 43% increase in CPU utilization and a 79% increase in response times (e.g., after time 350 seconds). Furthermore, in the Hard Reference case, major collections occur about four times as often as in the No Cache case.

As these results indicate, in-process caching can be an effective solution, provided that the system has sufficient memory available. In memory-constrained application environments, however, in-process caching can actually be detrimental to application performance. The primary reason for this is the impact of in-process caching on GC performance. It impacts GC performance in two important ways:

1. **Increases the frequency of GC:** Consider an application that has M bytes of memory available and consumes m bytes of memory per second when processing a certain number of requests. Suppose that when the available process memory falls below N bytes, the GC process runs. Then the frequency of GC is $(M-N)/m$ seconds. Now suppose that this application uses in-process caching, with C bytes of memory allocated to the cache. In this case, GC frequency will be $(M-$

CASE	EXPERIMENTAL RANGE (SECONDS)	METRIC	PERFORMANCE RESULTS
Soft Reference In-Process Cache vs. No Cache	0 - 349	CPU Utilization	Up to 36% reduction
	350 - 680	Response Time	Up to 14x reduction
		CPU Utilization	Up to 43% increase
Hard Reference In-Process Cache vs. No Cache	0 - 349	Response Time	Up to 38% increase
		Major collection frequency	3x increase
	350 +	CPU Utilization	Up to 43% increase
0 - 3600	Response Time	Up to 79% increase	
	Major collection frequency	4x increase	

TABLE 3: Performance comparison of in-process caching vs No Cache

ADVANTAGES:

- Provides fast access to objects
- Simple to implement

DISADVANTAGES:

- Consumes valuable application memory and CPU resources
- Increases GC costs
- Results in multiple copies of objects stored across application processes, which can cause poor memory utilization (low hit ratios) and risk of inconsistency among cached copies

TABLE 4: Advantages and disadvantages of in-process caching



authentic⁵

empowers business users to seamlessly capture both data and content directly into XML content through a visual interface, enabling transformation and reuse of corporate knowledge-

for FREE.

Capture Content into XML

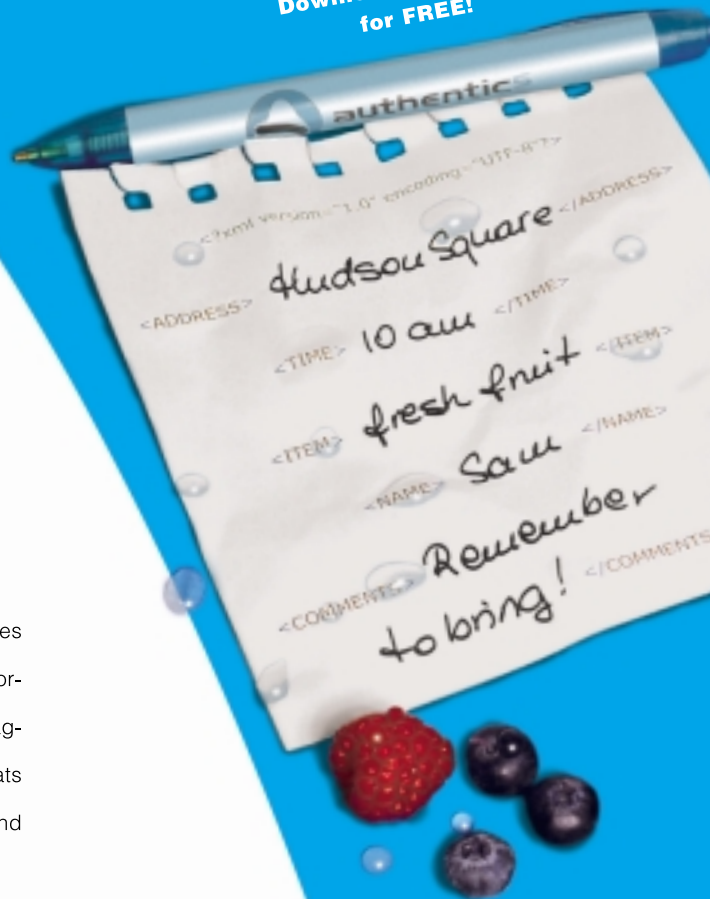
Business content is the lifeblood of knowledge companies - yet content producers waste excessive time locating and formatting content, instead of creating new content. Content fragmentation across workstations and proprietary file formats results in duplicated efforts, un reusable information assets, and is a major source of inefficiency.

authentic 5 is a light-weight, and easy-to-use XML document editor that integrates with all of the leading content repositories through standard interfaces. **authentic 5** features a browser plug-in, spell checking, real-time validation, and built-in templates for industry XML content formats like NewsML and DocBook. Customized user interface templates can be created for any XML content editing application.

authentic 5 gets business content into XML, ready for efficient storage and multi-channel publishing. As a free software license (yes, completely free), it's the most cost-effective solution for realizing standards-based, web-enabled, XML content editing, publishing and knowledge management applications! Learn more about how **authentic 5** can unlock volumes of information scattered across your organization today.

<http://jdj.altova.com/authentic5>

Download authentic 5
for FREE!



ALTOVA®

www.altova.com

CASE	EXPERIMENTAL RANGE (SECONDS)	METRIC	PERFORMANCE RESULTS
External Cache vs. No Cache	0 – 3600	CPU Utilization	Up to 21% reduction
		Response Time	Up to 71% reduction
		Major collection frequency	Approximately the same
External Cache vs. Soft Reference In-Process Cache	0 – 349	CPU Utilization	Up to 30% increase
	350 – 1040	Response Time	Up to 5x increase
		CPU Utilization	Up to 45% reduction
		Response Time	Up to 80% reduction
0 – 3600	Major collection frequency	70% reduction	
External Cache vs. Hard Reference In-Process Cache	0 – 349	CPU Utilization	Up to 30% increase
		Response Time	Up to 5x increase
	350 +	CPU Utilization	Up to 45% reduction
		Response Time	Up to 80% reduction
	0 – 3600	Major collection frequency	76% reduction

TABLE 5: Performance comparison of external caching vs other cases

$N-C/m$ seconds. Thus, the use of application memory for caching decreases effective available memory, which, in turn, causes GC to run more frequently.

2. **Increases the frequency of major collections:** Due to the high cost of major collections, it's preferable to have as few major collections as possible. However, it turns out that a major cost of in-process caching is a marked increase in the frequency of major collections. Objects remain in cache until they are evicted by either a replacement policy or an invalidation policy, at which time they are marked eligible for GC. Since cached objects tend to be long-lived, once evicted, they must be collected via a major collection. As a result, major collections occur more often.

To summarize, an in-process caching system can be an effective optimization solution for applications having sufficient memory available. However, in memory-constrained application environments, the in-process caching approach significantly increases GC costs. This, in turn, causes significant increases in application costs in terms of CPU utilization and response times.

Table 4 summarizes the advantages and disadvantages of in-process caching. Thus, the question that arises is whether it's possible to design a caching architecture that can still provide some of the advantages of in-process caching, but without the disadvantages.

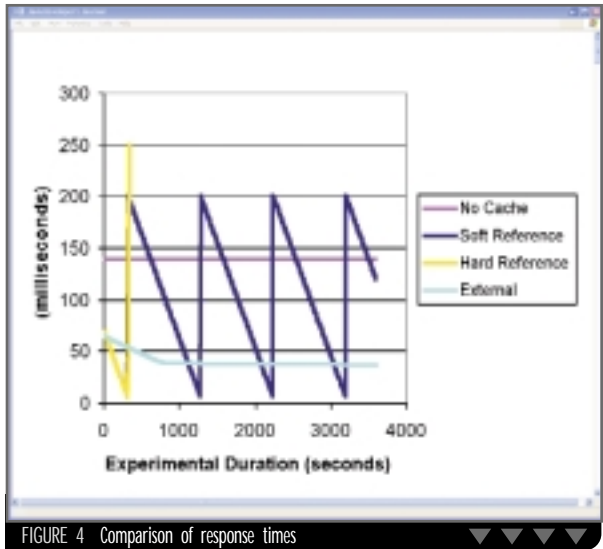


FIGURE 4 Comparison of response times

ADVANTAGES:

- Increases effective application memory and CPU resources
- Reduces GC costs
- Provides a single shared object storage, which improves memory utilization (higher hit ratios) and eliminates risk of inconsistent objects

DISADVANTAGES:

- Requires an interprocess call for each cache access
- Difficult to implement

TABLE 6: Advantages and disadvantages of external caching

An External Object Storage Solution

An external, centralized caching model for application objects is emerging as an alternative approach for optimizing applications. For example, the external servlet containers (SCs) architectural approach was described in a recent *JDJ* article by Mikhail Skorik ("External SC Architecture and VO Cache," [Vol. 7, issue 10]). Figure 5 depicts an external application object caching architecture, in which the cache is maintained and managed by a separate dedicated process, referred to as the cache server. The cache server is a single logical cache instance that can be shared by multiple application processes. Each application process communicates with the cache server through a lightweight communication library (shown as the cache client module in Figure 5) that is integrated with the application.

At runtime, this system works as follows. When an application creates an object instance that needs to be cached, it uses the communication library to serialize the object and store it in the cache server. When a subsequent request arrives for the same object instance, the application (through the communication library), checks the cache server for the object instance. If it's found, then it's served from the cache. Otherwise, the application creates the object instance and stores it in the cache server. All cache management functionalities, such as invalidation and cache replacement, are handled by the cache server.

To compare the performance of the external and in-process caching architectures, the same experiment described previously was run using Chutney Technologies' Aptimizer to store the profile objects externally. The Aptimizer runs on a separate Windows 2000 machine having a single 900MHz processor and 256MB RAM.

Figure 3D shows the CPU utilization for the External case. The CPU utilization is initially about 65%, and decreases over time until it reaches about 55%, where it remains for the remainder of the experiment. The response time results follow the same general pattern as the CPU utilization results. As shown in Figure 4, response time for the External cache is initially around 70ms. As the cache fills, less time is spent creating the profile objects, causing response time to decrease to about 40ms. At this point, the cache is full, so response time remains constant for the remainder of the experiment.

Table 5 provides a comparison of the External case with the other three cases for selected experimental ranges. With respect to the No Cache case, External caching provides up to a 21% reduction in CPU utilization, up to a 71% reduction in



PERFORMANCE IS EVERYTHING!

©2002 Precise Software Solutions, Ltd. All Rights Reserved.

Give your J2EE applications lightspeed performance.

To excel in today's business environment, your J2EE applications need to scream like a bat out of hell. With heavy traffic bottlenecks and crawling end user response times, you're total roadkill. Proactive detection of application performance problems is crucial to rapid resolution, and to taking the detour around a crisis up ahead.

Precise/Indepth for J2EE correlates web, J2EE and database server performance to get to the root cause of performance degradation, from URL to SQL and beyond™.

As Precise/Indepth for J2EE anticipates performance degradation, it instantly notifies your top mechanics for analysis and tuning deep within the middle tier of your web infrastructure and provides automatic drill-down analysis and recommendations. This translates into streamlined J2EE web applications and lightning fast response times.

Turn up the speed limit for your J2EE applications. Talk to your Precise account representative today at 1.800.310.4777, or visit www.precise.com/jdj to download a free white paper on J2EE performance management and to register for an upcoming Webinar on J2EE application performance.

Precise

Performance is Our Business



J2ME



J2SE



J2EE



Home

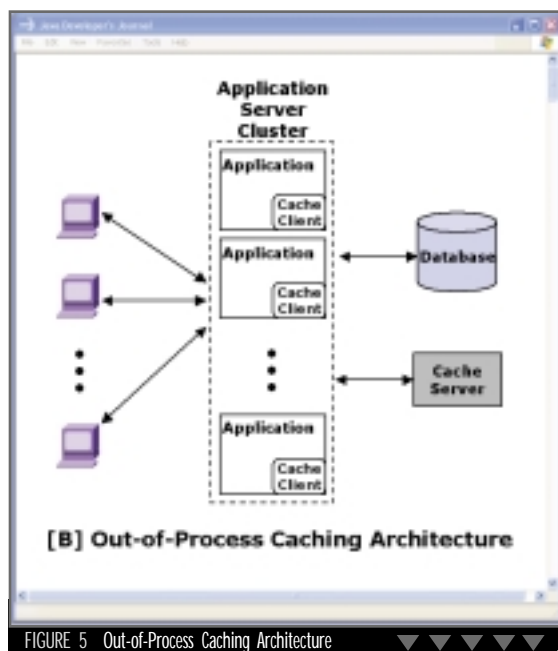


FIGURE 5 Out-of-Process Caching Architecture

response times, and results in no change in the frequency of major collections. These effects hold over the entire experimental range.

When compared to the in-process caching cases, the External case degrades performance initially (e.g., during time 0–349 seconds) in terms of CPU utilization (up to a 30% increase) and response times (up to a 5x increase). However, most of the time, the External case outperforms the in-process caching cases, providing up to a 45% reduction in CPU utilization and an 80% reduction in response times. This is somewhat surprising, given that the External case requires a network call for each access to the cache, whereas the in-process caching cases require only an in-memory lookup.

The improved performance of the External cache is due primarily to the fact that the External cache does not consume application memory for caching. Since the JVM memory utilization is low, the frequency of GC – and of major collections in particular – is quite low. Notably, the External case provides

ing an effective external caching system is that it requires an interprocess call (and perhaps a network call) between the cache server and the application process for each access to the cache server. Commercial solutions, such as main-memory databases (e.g., TimesTen), object databases (e.g., Excelon's ObjectStore), and other object storage solutions (e.g., Chutney's Aptomizer), are designed to meet the stringent throughput and response time requirements for an external caching architecture.

Conclusion

In this article, I have shown that the widely used in-process caching approach is not always the ideal optimization solution for enterprise applications. For applications having modest memory requirements, an in-process caching system may be an effective optimization solution. Many enterprise applications, however, do not have such modest memory requirements. For these types of applications, in-process caching can significantly increase GC costs, which, in turn, causes significant increases in application costs in terms of CPU utilization and response times.

An alternative optimization solution that deserves consideration is an external caching architecture. An external cache can increase the effective memory available to the application, thereby reducing the frequency of GC and of major collections. This reduction in GC costs can lead to significant improvements in application performance in terms of CPU utilization and response times over an in-process caching solution. ☛

References

- *BEA Systems WebLogic JRockit*: www.bea.com/products/weblogic/jrockit
- *Chutney Technologies, Chutney Aptomizer*: www.chutneytech.com/products
- *Excelon Corp. ObjectStore*: www.exln.com/products/objectstore/
- *JSR #107: JCACHE – Java Temporary Caching API*: <http://jcp.org/en/jsr/detail?id=107>
- Pawlan, M. (1998). *Reference Objects and Garbage Collection*: <http://developer.java.sun.com/developer/technicalArticles/ALT/RefObj>

“An external cache can increase the effective memory available to the application, thereby reducing the frequency of GC and of major collections”

a 70% reduction in major collection frequency when compared to the Soft Reference case and a 76% reduction when compared to the Hard Reference case. The end result is that the External case spends less CPU time in GC and, therefore, spends more CPU time in application processing.

To summarize, the external caching architecture has several advantages over the in-process caching architecture, as shown in Table 6. By storing objects out-of-process, external caching allows application memory to be dedicated for application processing. As a result, GC is invoked less frequently, reducing the CPU time spent in GC for the application. Moreover, the number of long-lived objects in application memory is reduced, which reduces major collection frequency.

There are also disadvantages associated with external caching (see Table 6). Perhaps the greatest challenge in build-

- Sun Microsystems. “Tuning Garbage Collection with the 1.3.1 Java Virtual Machine”: <http://java.sun.com/docs/hotspot/gc>
- Sun Microsystems. *Java HotSpot Technology*: <http://java.sun.com/products/hotspot>
- *TimesTen Software*: www.timesten.com/products

AUTHOR BIO

Helen Thomas is a cofounder of Chutney Technologies and a technical expert in the area of decision-support databases. She is also an assistant professor of information systems at Carnegie Mellon University. Helen received her PhD in information technology management from Georgia Tech, an MSE in operations research/industrial engineering from the University of Texas at Austin, and a BS in decision and information sciences from the University of Maryland at College Park.

helen.thomas@chutneytech.com

Rational XDE Professional Java Edition is:

UML modeling for Java

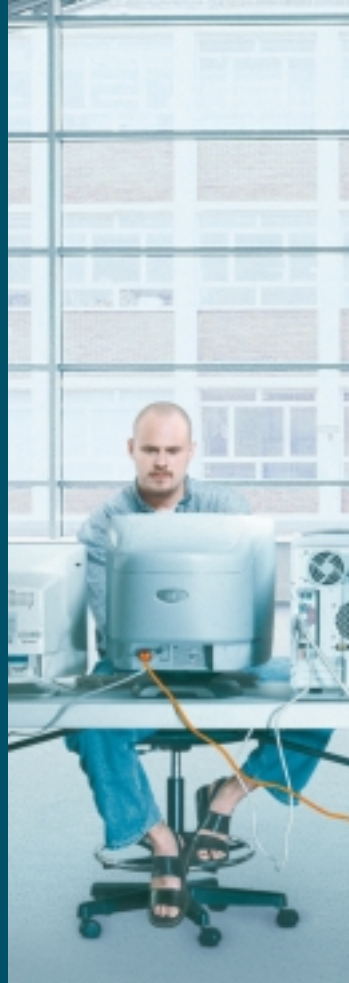
model ↔ code synchronization

automated design documentation

customizable design patterns

data modeling

and runs inside WebSphere Studio



**In other words,
your IDE on steroids.**

Lemme guess. You're thinking this is just one more tool that's actually going to get in your way. Hardly. Rational® XDE™ Professional blends right into your development experience. How? It runs inside IBM® WebSphere® Studio Application Developer, or stands alone with its own Eclipse-based Java™ Integrated Development Environment. It supports J2EE and J2SE. And Rational XDE support also includes membership to the Rational Developer Network,SM a helpful online community that provides reusable assets, Web-based training and discussion forums. We figure if someone else has already solved a problem, why should you have to do it again? To create a better development experience without creating new problems for yourself, get Rational XDE starting at only \$1,595. Or go to www.rational.com/offer/javacd2 to get a free Rational XDE Professional Evaluation CD so you can see it and try it out for yourself.

be liberated

Rational[®]
the software development company

Rational and the Rational logo, XDE and Rational Developer Network are trademarks, registered trademarks or service marks of Rational Software Corporation in the United States and in other countries. IBM and WebSphere are registered trademarks of IBM Corporation. Java is a trademark of Sun Microsystems Inc. in the United States and in other countries. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective companies. ALL RIGHTS RESERVED. ©2003 Rational Software Corporation.

Taking Continuous Integration to the Next Level

Continuous what?



WRITTEN BY
FRANZ GARSOMBKE

Sometimes, the easiest and most rewarding part of development is the actual coding. Managers and developers often dismiss the building, deploying, testing, and metrics-gathering aspects of the software life cycle.

Continuous integration is the concept of automating the build cycle so that code is built and tested many times during the day. This allows developers to integrate their code into the build daily or hourly, thus lowering the chances of integration problems.

In this article, I introduce an Ant-based build and integration system called BigBrother that leverages XP (Extreme Programming) principles. BigBrother combines several open-source tools into a “continuous integration” system. I also offer a real-world scenario in which BigBrother was used to successfully manage a continuous build process.

The concept of continuous integration has been around for a long time and one of its biggest proponents is Extreme Programming. XP is a disciplined approach to software development that stresses communication, simplicity, feedback, and iterative design. One of the main coding practices of XP is to integrate often. The benefits of integrating code are numerous:

- Developers are guaranteed to be working with the latest version of software.
- Compatibility problems are more likely to be detected and avoided early.
- Integration cycles that used to take days or weeks are reduced to minutes.
- Communication between developers increases.
- Integration bugs are found much sooner than later.

Automating the build-and-test cycle definitely brings many benefits to any software development life cycle, but this only scratches the surface of what can be done by combining some of the open-source tools available today.

Paradise Found

With the advent of Java and open-source projects, creating and automating a continuously integrated environment has become simple, yet extremely powerful. These open-source projects include components for compiling and deploying code, executing automated unit tests, computing and displaying unit test code coverage, calculating various code metrics, as well as many other tools for making a project successful. All of these individual components can then be wrapped and automated with the help of yet other open-source projects.

Using each of these tools by them-

selves is helpful in the development of software, but the real power is when you combine them into a single continuous integration environment that compiles, tests, and reports metrics on a code base every time a file is checked in. Imagine a development environment with the following features:

- Project Web site with up-to-the-minute build information and code metrics.
- Development team is notified of broken builds immediately and accountability is placed on the last person(s) to check in code.
- Unit tests for the entire project are run automatically after every code

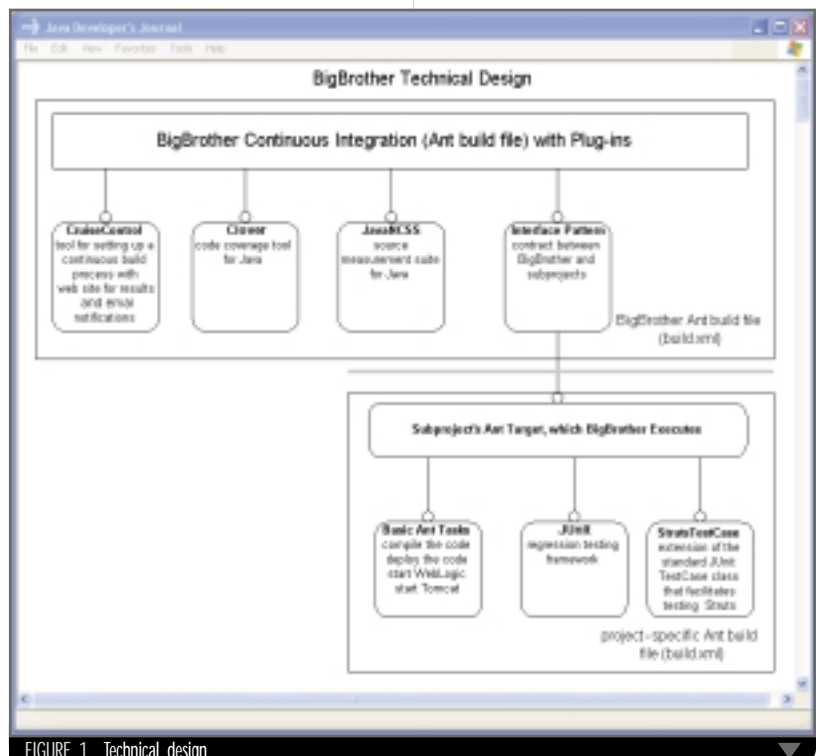


FIGURE 1 Technical design

Like most developers, you're tired of hearing "more productivity." So we disguised it as a chart.

Crystal Reports 9 Advanced Edition Product Features				
POWER	Report Designer - Use intuitive visual interfaces and Wizards to create highly formatted reports without any programming	✓	* Custom Data Access - Access in-memory data through JavaBeans	✓
	Report Integration - Integrate reports into web, wireless, and windows applications using Java, .NET, and COM SDKs	✓	100% Java SDK - Deliver reports in Java applications within a J2EE environment	✓
	* Data Access - Access virtually any data with native, ODBC, OLE DB, and JDBC connectivity to relational, OLAP, XML, legacy, and enterprise data sources	✓	Advanced Parameters - Deliver reports with pre-defined parameters for processing unique report data views at runtime	✓
	Web Report Viewers - Deploy reports over the web with interactive, zero client viewer controls	✓	Report Creation and Modification APIs - Use these rich APIs to extend runtime report manipulation and customization to end users	✓
	* Custom Report Style Templates - Eliminate redundant formatting by creating and applying custom templates across multiple reports	✓	* Unicode Support - Display data stored in virtually any language in a single report	✓
	* Repository - Centrally store key report components in the Repository for sharing across multiple reports and single point updating	✓	* Formula Workshop - Use Crystal Reports' extensive formula language for complete, control over report formatting, complex logic, and data selection	✓
	* Report Parts - Specify existing key report elements, Report Parts, for access via wireless applications and tight integration with web applications	✓	Guided Navigation - Establish seamless navigation paths between report objects for an improved end user experience	✓
	Java Viewer - Render reports to pure HTML or WML for mobile clients with a server-based Java report viewer	✓	Exports - Integrate multiple export options including XML, PDF, DHTML, Excel, Word, RTF, Text & email	✓
	FLEXIBILITY	EASE OF USE		

* New or updated features



Crystal Reports is an 11th generation high-productivity reporting solution for tightly integrating dynamic content into enterprise web applications using Java, .NET, and COM. Allowing detailed control of end user report viewing, interaction, and runtime modification, Crystal Reports is also a fundamental component of the Crystal Enterprise web-based information delivery system. A world leader in reporting, chosen by technology partners including Microsoft and SAP, Crystal Reports is also supported by a full range of online and onsite training and technical support.

To learn more visit www.crystaldecisions.com/cr9/218 for free online seminars, demos, and evaluation software. To purchase Crystal Reports, Call 1-800-877-2340, visit our online store at www.crystaldecisions.com/store/218, or contact your reseller.

© 2003 Crystal Decisions, Inc. All rights reserved. Crystal Decisions, Crystal Reports, Crystal Enterprise, and the Crystal Decisions Logo are either a trademark or registered trademark of Crystal Decisions, Inc. in the U.S and/or other countries. All other trademarks or registered trademarks referenced are the property of their respective owners.

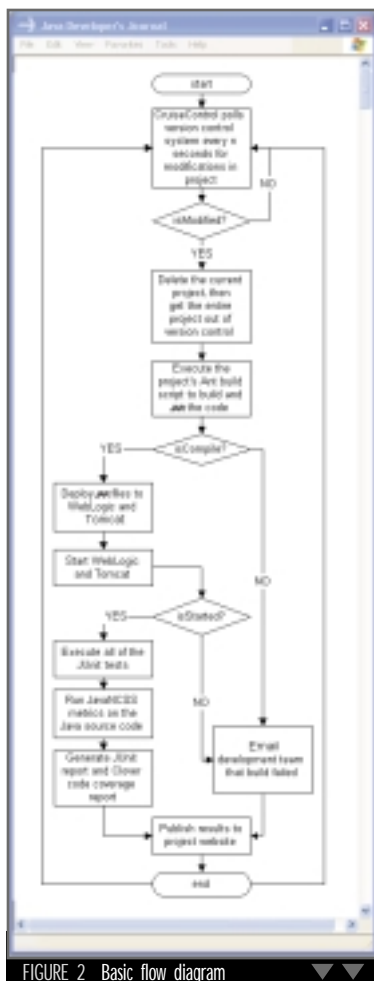


FIGURE 2 Basic flow diagram

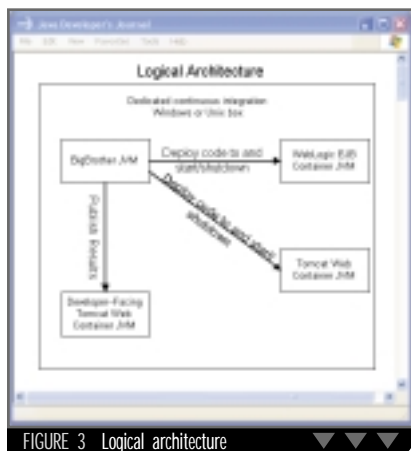


FIGURE 3 Logical architecture

change. Results of these tests are reported on the project Web site.

- Code metrics are run after every code change, showing such things as Javadoc coverage, cyclomatic complexity of each method, and lines of code.
- Code coverage report displaying how much of the code was tested by the automated unit tests.
- Provide team leads with the ability to monitor coding and testing efforts on an up-to-the-minute basis.

Murphy's Law states that a developer would check code in on Friday and break the build without anyone knowing it. Many weekends were lost due to bad builds

- Both Web-based container and EJB container code can be fully and automatically unit tested.
- Reporting on code metrics can be done on both the client and server tiers.
- XP-style feedback – every build is a snapshot of how much of the project has been completed and how much has been unit tested.

This might seem like a lot to ask for on every Java project you undertake, but using Apache's Ant as the glue makes it easy to tie various open-source projects together such as CruiseControl, Clover, JavaNCSS, JUnit, and StrutsTestCase for JUnit. I've combined all these components together into a system affectionately called BigBrother. BigBrother is simply an Ant build process that pulls together all the open-source tools listed above into a single integrated environment. A development team using this environment is forced to monitor all the code they check in and to make sure it compiles, doesn't break any unit tests, and is well tested...because BigBrother is watching.

The Tools in the Toolbox

Before I explain how BigBrother works, I need to explain the various components used to build the continuous integration environment. Apache's Ant is the cornerstone of this architecture. Ant is a Java-based framework that has revolutionized the build process. Ant has many built-in tasks that can be used for everything from compiling and packaging code, to starting and stopping application servers. Ant can also be easily extended using Java classes to provide customized functionality. Ant is the perfect tool for tying many disparate components together because of its flexible, task-based architecture.

CruiseControl, open sourced by ThoughtWorks, is the second largest piece in the BigBrother puzzle. This pluggable open-source framework allows for a continuous build process and includes a Web interface to view the

details of the current and previous builds. It has built-in version control polling for code changes and e-mail notifications for builds that are not successful.

Clover is a code coverage tool for Java and the only commercial software in the BigBrother architecture. While Clover is not open source, it's very inexpensive given the benefits derived from its use. Code coverage is used to determine which sections of code are not being executed by the existing set of unit tests. Using a code coverage tool exposes exactly how well developers are testing their code down to the method, statement, and branch level. It also helps determine a quantitative measure of code coverage. This measure has been theorized as an indirect correlation to the quality of code delivered.

JUnit is a Java framework for creating repeatable automated unit test cases. Some features include test fixtures for sharing common data, test suites for organizing tests, and methods for testing expected results. JUnit is great for testing most Java classes, but there are times when you'll need something more robust. There are many extensions for JUnit, ranging from the ability to unit test Web-based code to the ability to unit test J2ME code.

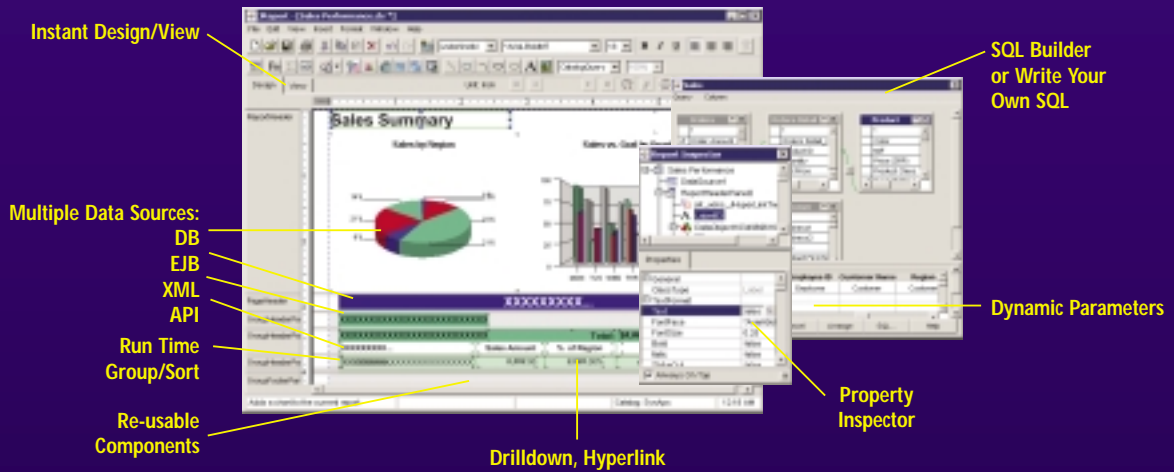
JavaNCSS is a source measurement suite for Java. It provides helpful metrics to gauge how well code is being programmed and commented. Some of the metrics are cyclomatic complexity per method (program complexity number); noncommenting source statements; number of formal Javadoc comments per class and method; and the number of packages, classes, and functions.

Combining all these components into one build environment allows any development shop, large or small, to have a world-class enterprise build solution.

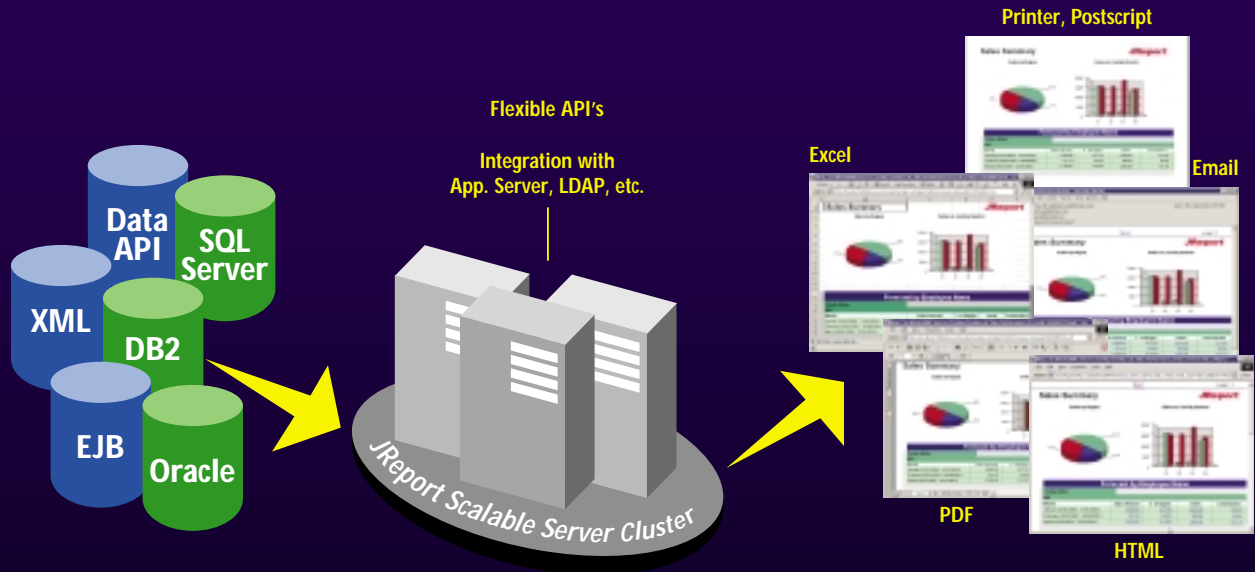
BigBrother 24/7

Now that all of the different tools have been defined, let's explore how they fit into the overall picture.

DESIGN ONCE...



REPORT ANYWHERE.



JREPORT™

JReport lets you take your reports anywhere you want. Imagine creating reports that pull together data from virtually anywhere, arranging them any way you wish, and delivering them precisely – whether in pure HTML, PDF, Excel, XML, JSP, email, or your network printer. Imagine having ad hoc reporting ability to create or modify reports on the fly, right from the end user's browser.

And all this is built on 100% J2EE architecture, to provide secure, reliable, scalable performance and easy integration with your enterprise systems. You can count on JReport: any report, any format ... anywhere.



©COPYRIGHT 2002 JINFONET SOFTWARE, INC. ALL RIGHTS RESERVED. JREPORT AND JINFONET ARE TRADEMARKS BELONGING TO THEIR RESPECTIVE HOLDERS.



J2ME



J2SE



J2EE



Home

Figure 1 depicts the components that make up BigBrother. The top rectangle signifies the BigBrother Ant build file. The Ant file can be seen as both the wrapper and the driver for the continuous integration cycle. This build file has tasks that control the different components that have been plugged in. The BigBrother build environment can control multiple subprojects at once. The interface design pattern is implemented to execute a subproject's specific Ant target.

The bottom rectangle symbolizes the subproject's build file. BigBrother can concentrate on monitoring the automated build cycle while leaving all the nongeneric work, like compiling, deploying, starting, and stopping the different application servers, and unit testing code, to the subproject.

There exists a contract between BigBrother and each of its subprojects. Each subproject implements an Ant target that BigBrother has been told to call ahead of time. This specific target will perform all the necessary work needed to compile, deploy, and test the subproject. BigBrother is blind to the functionality in its subprojects, caring only about executing the correct commands. Using this design, you need to write only one "master" BigBrother build file, as it's reusable across all the subprojects.

As shown in Figure 2, there's a rigid and repeated structure to all BigBrother software builds. After configuring

CruiseControl to poll every n seconds and starting the CruiseControl Java Virtual Machine (JVM) using Ant, the project build cycle is literally on "cruise control." If modifications have been found in the code base, a fresh copy of the entire branch is retrieved out of the version control system; the code is compiled, deployed, and then unit tested on the appropriate application servers. Code coverage and metrics reports are then generated from the code branch and the results are published to a Web site accessible by developers and project managers. If at any time in the process the build fails, all the developers on the project are sent an e-mail containing the location of the Web site for details about why it failed. There's also an option to spam developers if the build stays broken. Needless to say, this is not a popular option.

Figure 3 shows that the logical architecture is made up of a dedicated machine to perform the continuous integration tasks. This machine has everything it needs to build, deploy, and test an entire J2EE environment. It consists of a WebLogic application server used to deploy the server-side code, Tomcat Web container used to deploy the client-side code, and another Tomcat Web container used to publish the results after the build process is complete. Multiple instances of BigBrother can run on the same dedicated machine, thereby continuously inte-

grating multiple projects. It also takes advantage of having only one instance of WebLogic running multiple domains and one instance of Tomcat running different ports for its unit testing.

The Rules

Certain rules must be put into place as part of the software development life cycle when using BigBrother. The first rule of BigBrother is do not check in code that does not compile. The second rule is do not check in code that does not compile. The third rule is that new code checked into version control cannot break any of the existing unit tests. The fourth rule is that any new code checked in must have working unit tests. The last rule is that any new code checked in must have at least a 90% code coverage rating after running its unit test(s). The rule "traffic cop" in this case is BigBrother. If any of these rules are broken, BigBrother will be the first to know and will "punish" accordingly (usually a friendly e-mail reminder of the offense). Adhering to these rules will greatly improve the chances of a zero-length integration life cycle and a much shorter quality assurance (QA) life cycle.

The Real World

If, after reading all of the above, you're still not a believer in having a continuous integration environment, here's a real-world case study to ponder. Before BigBrother, my company's soft-

AUTHOR BIO

Franz Garsombke, a senior software engineer at Global Healthcare Exchange, has been developing and architecting enterprise software solutions for the past seven years, using Java for the past four years. Franz is a huge proponent of open source code and frameworks.

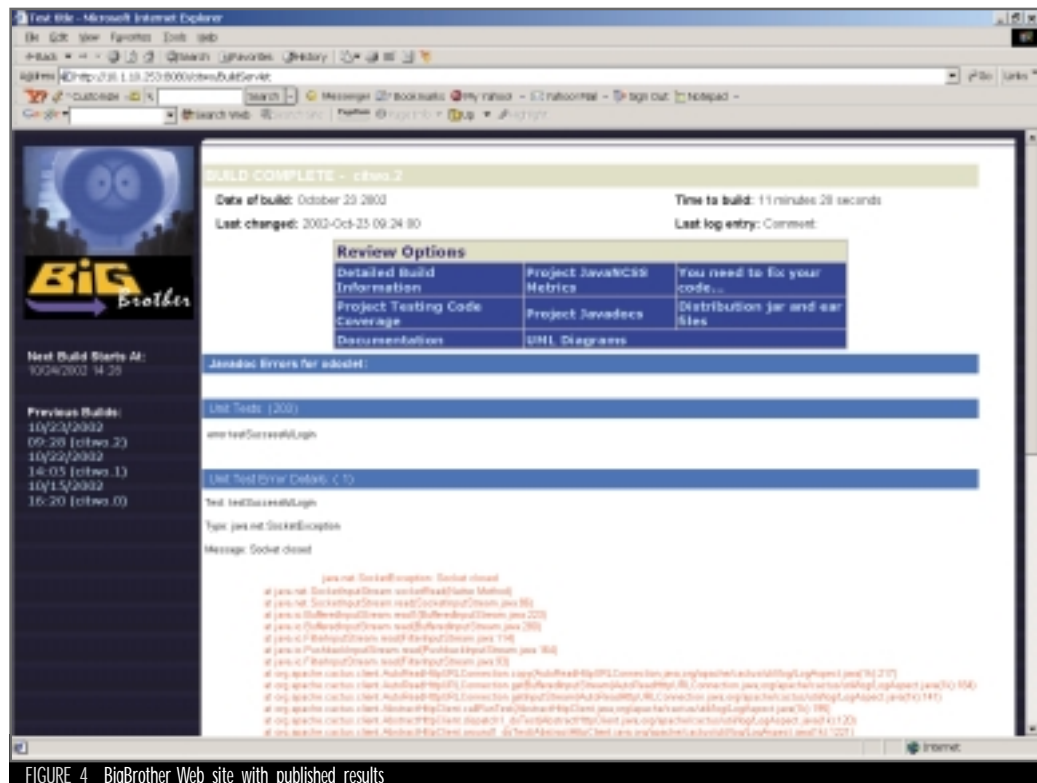


FIGURE 4 BigBrother Web site with published results

ware builds were in a state of total disarray. If you were lucky enough to check out a branch of code that compiled, you still weren't guaranteed that the code actually worked since there were no unit tests. Developers had to wait hours for code to be fixed so that the build would compile.

Murphy's Law states that a developer would check code in on Friday and break the build without anyone knowing it. Many weekends were lost due to bad builds. Nobody knew how many lines of code there were in a certain project or the complexity of the methods and classes. Nobody knew how much of the code had Javadoc in it. A striking figure was that only 65% of the code moved to QA actually worked. Development life cycles were greatly extended because of code integration issues.

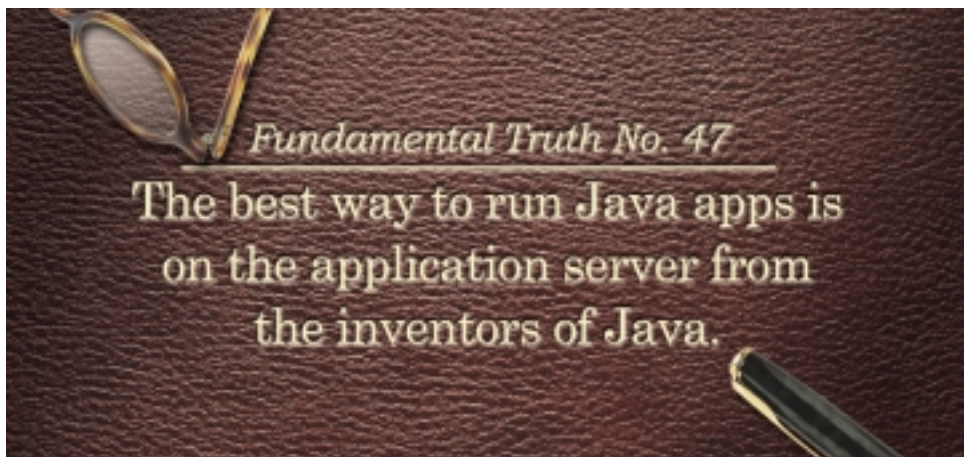
After implementing BigBrother across the board on all of my company's development projects, a dramatic shift occurred. The builds are seldom broken; and if a build is broken, it's fixed in a matter of minutes. Developers now take pride in their work and a competitive spirit has arisen to see who can have the most code coverage out of their unit tests. Every functional piece of code checked in now has a corresponding unit test. The result that managers love is now, on average, 97% of the code moved to QA is bug-free.

Integrate Often

A continuous integration environment literally turns software code into an open book. (see Figure 4). Detailed information is provided on who has been doing their unit testing, who has been checking in code that compiles, and who has been creating simple yet functional code. It also allows communication channels to open up more quickly between developers, and generates more of a team-oriented approach rather than the lone developer. Generated reports do not lie; they paint a picture of the current status of a project. The investment time up front to automate the build and test process is so slight and the rewards so beneficial that no development shop should be without a continuous build environment.

References

- *Continuous Integration:* www.martinfowler.com/articles/continuousIntegration.htm
- *Extreme Programming:* www.extremeprogramming.org/index.htm
- *Cruise Control:* <http://cruisecontrol.sourceforge.net>
- *Apache Ant:* <http://jakarta.apache.org/ant>
- *Clover:* www.thecortex.net/clover
- *JUnit:* <http://junit.sourceforge.net>
- *StrutsTestCase for JUnit:* <http://strutstestcase.sourceforge.net/>
- *JavaNCSS:* www.kclee.com/clemens/java/javancss
- *Struts:* <http://jakarta.apache.org/struts>



Open. Scalable. Reliable.

Looking to develop Web services quickly and economically? Sun™ ONE Application Server is designed to work with Sun ONE Studio for Java™—giving you full Java IDE capabilities. You can develop applications in Solaris™, Linux®, or Windows® environments and easily deploy to Sun ONE Application Server 7.

Sun ONE Application Server 7 has a reputation for being:

- Easy to install
- Easy to configure
- Easy for systems administrators to monitor

Try it now! Evaluate both the Sun ONE Application Server 7 Platform Edition and/or the Standard Edition with Sun ONE Studio for FREE*. Download it today at www.intraware.com/evalapp/

Plus, with every evaluation, you'll receive a complimentary copy of "Designing Enterprise Applications with the J2EE Platform"—valued at \$45!

For more information about these or other innovative solutions from Software Spectrum or Intraware, call **1-800-469-4313** or email sales-sunone@intraware.com

*Free software download is only available in the U.S. Offer ends June 30, 2003.





JASON BELL J2SE EDITOR

Train-ing

Training – it can be a dirty word to some; learning by doing is all very well but what do you actually learn?

On my daily commute to work, I have 90 minutes on the train to read, digest, and think about how to implement these new practices into my programming. I don't know how much audio training exists for Java, but if you can listen to tapes in the car, you have, in effect, a mobile library of information (I learned this from Brian Tracey, a sales mentor; getting into the brain of a salesman is really interesting, and it also means you can play their game a bit when they suggest new features for your software). Preparation is also key, and reading on the train or listening to tapes will help. I know a lot of you spend many out-of-work hours brushing up on Java, design patterns, and everything else the industry sends our way.

I've learned more from books than from blogs, as the latter tend to be based more on opinions than anything else. Blogs are great for catching up on what's going on with recently released software and libraries. If you want to seriously start learning though, you'll have to dig deep and be prepared to sacrifice some time. Let me tell you a short story by first asking a question.

How well do you know your compiler? It sounds like a silly question, doesn't it? But I think only about 40% of Java developers know of the Javac compiler options. To be honest, we try to get away with using only what we have to and don't bother with the rest. If you are using an IDE, you may never really know what is going on under the bonnet.

What got me into all this? Bruce Eckel, yes, it's all his fault! Traveling on the train, I started reading odds and ends from *Thinking in Java* (3rd edition) and stumbled on memory profiling. I don't have a huge interest in the subject, though I have

to admit that I'm currently using Rational Purify Plus for a future review in *JDJ* so I didn't even consider that the SDK would have one. My eyes lit up when I saw:

```
java -
Xrunhprof:heap=sites,cpu=samples,depth=1
0,monitor=y,thread=y,doe=y
ClassToProfile
```

It's worth noting that the *X* options are nonstandard and liable to change, but still worth investigating. I had a quick look at Jikes as well, and build 1.17 didn't have it. Running `-Xrunhprof` is an adventure in itself (searching Google will produce some more meaningful information, as will the Sun site). The memory profiler produces a text file with a lot of information about thread usage and which objects are hogging all the allocated memory. The only downside is that in the commuter trains everyone within a meter radius is reading the finer art of object inheritance and memory profiling.

Another aspect of learning is tutoring or one-to-one help. We've all had times when we thought we knew something, then we met someone who basically blows our ideas out of the water. It happened to me not so long ago; in fact the person in question has written a couple of times for *JDJ*. He got me into some basic design patterns, especially the use of the Factory pattern and using abstract classes as interfaces; this actually changed the way I program. What was in error was my logic and thought process. It's worth taking the time to talk, read, and, most important, listen to your peers. Your way of doing something may be correct, but I can guarantee that there are three other ways to do what you're doing. Try them all and learn from them, good points and bad.

Right, I'm off to find a new book to take on the train with me. Something on design patterns to get Alan fired up. ☘

▼▼▼ jasonbell@sys-con.com

AUTHOR BIO

Jason Bell is a programmer and chief technical officer for a B2B Web portal in York, England. He has been involved in numerous Web projects over the past five years, the last two of which have been servlet-based.

Train-ing

Training – it can be dirty word to some; learning by doing is all very well but what do you actually learn?

On my daily commute to work, I have 90 minutes on the train to read, digest, and think about how to implement these new practices into my programming. I don't know how much audio training exists for Java, but if you can listen to tapes in the car, you have, in effect, a mobile library of information.

by Jason Bell

36

SWT – A Native Widget Toolkit for Java

The Standard Widget Toolkit (SWT) is a Java class library that allows you to create native user interfaces. It's designed to provide efficient, portable access to the underlying facilities of the operating system on which it's implemented. SWT uses native widgets wherever possible, giving an SWT program a native look and feel and a high level of integration with the desktop. In addition, SWT includes a rich set of controls such as a tree, table, and tab folder. This article introduces SWT by describing some of the basic concepts and classes.

by Steve Northover and Joe Winchester

38

SWT

A NATIVE WIDGET TOOLKIT FOR JAVA

WRITTEN BY STEVE NORTHOVER AND JOE WINCHESTER

PART 1 OF 2

ACHIEVE
A HIGH LEVEL
OF JAVA CLIENT
INTEGRATION
WITH THE
DESKTOP OS

THE STANDARD WIDGET TOOLKIT (SWT) IS A JAVA CLASS LIBRARY THAT ALLOWS YOU TO CREATE NATIVE USER INTERFACES. IT'S DESIGNED TO PROVIDE EFFICIENT, PORTABLE ACCESS TO THE UNDERLYING FACILITIES OF THE OPERATING SYSTEM ON WHICH IT'S IMPLEMENTED. SWT USES NATIVE WIDGETS WHEREVER POSSIBLE, GIVING AN SWT PROGRAM A NATIVE LOOK AND FEEL AND A HIGH LEVEL OF INTEGRATION WITH THE DESKTOP. IN ADDITION, SWT INCLUDES A RICH SET OF CONTROLS SUCH AS A TREE, TABLE, AND TAB FOLDER. THIS ARTICLE INTRODUCES SWT BY DESCRIBING SOME OF THE BASIC CONCEPTS AND CLASSES.



Hello World: A Simple SWT Program

The easiest way to learn SWT is to study a simple example. The following code shows a complete SWT program that creates and displays a new window on the desktop with “Hello World” in the title bar. Figure 1 shows the result of running this program on Windows XP.

```
1 import org.eclipse.swt.*;
2 import org.eclipse.swt.graphics.*;
3 import org.eclipse.swt.widgets.*;

4 public class HelloWorld {
5     public static void main(String[] args){
6         Display display = new Display();
7         Shell shell = new Shell(display);
8         shell.setText("Hello World");
9         shell.setSize(200,100);
10        shell.open ();
11        while (!shell.isDisposed()) {
12            if (!display.readAndDispatch())
13                display.sleep ();
14        }
15        display.dispose ();
16    }
17 }
```

Lines 1-3: SWT is divided into a number of packages, each modeling a different area of the user interface. All packages begin with the prefix `org.eclipse.swt`. The package `org.eclipse.swt` itself contains classes used by every package in SWT, such as constants and exceptions, while `org.eclipse.swt.widgets` contains only the widget classes. We didn't actually need to import `org.eclipse.swt` itself or `org.eclipse.swt.graphics` in this example because we didn't use any classes from these packages. However, any significant SWT program will use these three packages and this is a good place to introduce them.

Line 6: Every SWT program must create a `Display`. Typically, the first line of an SWT program creates a display, which SWT uses to establish the connection with the underlying platform window system. The class `Display` is very important in SWT. For readers familiar with the X Windows system, an SWT `Display` is equivalent to an X Windows `Display`.

Lines 7-9: These lines create a shell, and set the title and the size. Top-level windows are represented by the class `Shell` and are created on a `Display`. It's not necessary to set either the title or the size of a shell when it's created. A shell can have an empty title and the window system will assign the initial position and size using the standard algorithm of the desktop.

Line 10: A shell is invisible when it's created. The method `open()` makes the shell visible, brings it to the front on the desktop, and sets user input so when keys are typed, they go to the shell.

Lines 11-14: Like most modern window systems, SWT supports an event-driven user interface. This requires an explicit event loop that repeatedly reads and dispatches the next user interface event from the operating system. When there are no more events to dispatch, the program goes to sleep waiting for the next event, yielding the CPU to other programs on the desktop. The loop terminates when the programmer decides that the program has ended, typically when the application's main window, in our case the shell, closes. It's up to the application programmer to decide the exit condition for the event loop, as only he or she knows when the program is finished.

Line 15: This line disposes the `Display`. Strictly speaking, it's not necessary to dispose the `Display` as long as the program

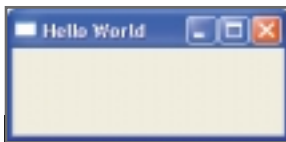


FIGURE 1 Hello World on Windows XP

exits to the operating system. In all modern operating systems and on every platform SWT supports, exiting to the operating system releases any resources that were acquired by the process, including the resources acquired by SWT through

to have an arbitrary depth. When the child of a shell is another shell, the child is commonly called a dialog shell. A dialog shell floats on top of its parent shell.

It's important to understand the difference between the runtime and inheritance hierarchy of controls. The hierarchies constructed at runtime are containment hierarchies. A Shell is not a subclass of Display, but a display can contain a list of shells. Figure 2 shows the inheritance hierarchy of Display, Shell, Button, and Group, along with the containment hierarchy that's built at runtime from instances of these classes.

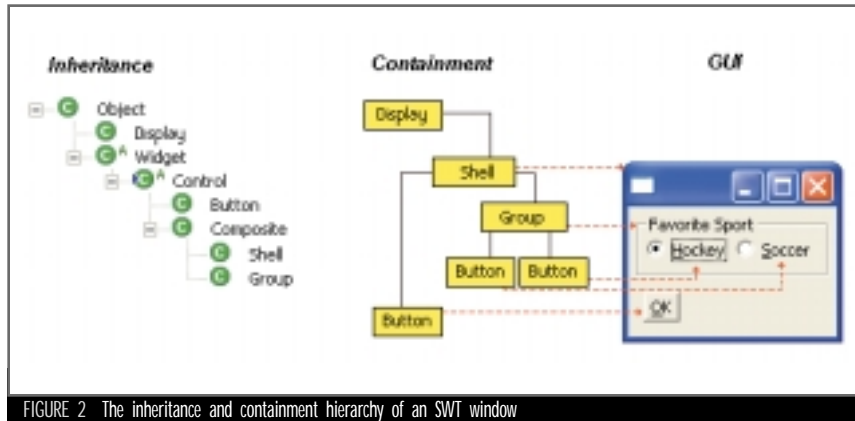


FIGURE 2 The inheritance and containment hierarchy of an SWT window

The concept of a containment hierarchy is an important one in SWT, so much so that it's reflected in the constructor of almost every object.

Controls are created using a constructor that takes the parent and a style. Styles are bit values used to represent operating system features that are create-only, such as multi- or single-line text edit capability, scroll bars, or borders. Because these operating-system features cannot be changed after creation, the style of a

control cannot be altered after instantiation. Style bits provide a compact and efficient method of describing the create-only properties of a control. An alternative implementation would be a large number of constructors in each class containing an arbitrary list of boolean parameters, each parameter in an arbitrary order.

As expected, you can combine styles by using a bitwise-or operation. All styles are defined as constants in the class org.eclipse.swt.SWT. For example, the following code fragment creates a multiline text control that has a border and horizontal and vertical scroll bars:

```
new Text(composite, SWT.MULTI |
        SWT.V_SCROLL | SWT.H_SCROLL |
        SWT.BORDER);
```

To use the default style of a control, the style value SWT.NONE is specified in the constructor. The list of the style constants that correspond to each control is described in the documentation for the control.

A list of the basic SWT controls is shown in Table 1.

Events

So far we've described how to create controls and set various properties, configuring them either in the constructor or through methods that are public APIs. These are all operations performed on a control by the programmer. However, controls are also visual user-interface elements. What happens when the user presses a button or selects an item from a list?

We have already described the display and its role in the implementation of the event loop, reading and dispatching events from the operating system. In fact, without an event loop, an SWT program will terminate and return to the operating system when main() terminates. The event loop plays a critical role interacting with the user. Events are generated by the user, dispatched by the event loop, and application code is invoked in response. Application code regis-

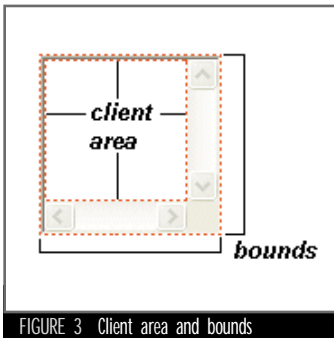


FIGURE 3 Client area and bounds

the Display. So why include this line in the sample program? Not only is it good form to dispose of resources that you acquire, it can help you find places where you are not explicitly disposing resources, especially when using third-party or operating system-level tools that look for leaks.

As long as the window remains open, the event loop keeps Java alive and the SWT program keeps running.

Widgets: Shells and Controls

A Control is a user-interface element that's contained in a shell. Controls are common in all user interfaces. Buttons, labels, trees, and tables are all controls and users are familiar with these from other programs on the desktop.

Taking a bottom-up view of the world, every control has a parent and this parent can be another control, called a Composite. Shell is a subclass of Composite, so shells can have children. The parent of a shell is the display or another shell. Stated another way, this time from the top down, a display contains a list of top-level shells, where each shell is the root of a tree containing subtrees composed of composites and controls. Composites contain other composites, allowing the tree

CONTROL	DESCRIPTION
Button	A push, check, radio, toggle or arrow button SWT does not have a separate class for each kind of button.
Label	A label showing a string, image or separator (a vertical or horizontal line)
Text	A single or multi-line text edit control
Slider	A control that represents a range of numeric values
ProgressBar	A control used to indicate percentage completion of a task
Sash	A selectable object to allow dragging of a rubber band area
Scale	A control that represents a range of numeric values with annotation marks
Combo	A control that gathers input from a dropdown list or by direct text entry
List	A control that allows single or multi-selection from a list of strings
Composite	A control that allows grouping of other controls
Group	A composite with a title and border, whose children are often radio buttons
Shell	A top-level window that has a title and window manager decorations
Canvas	A control designed to allow arbitrary graphics operations

TABLE 1: Basic SWT controls

The **only** Java components you need for J2EE or Swing development

JClass[®]

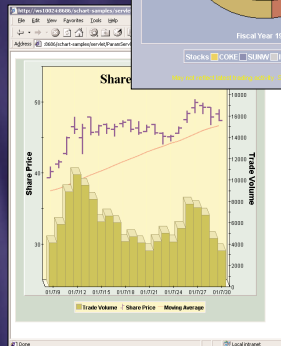
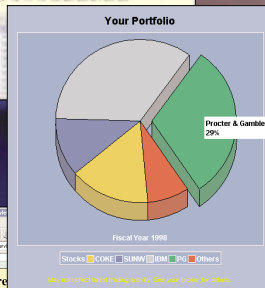
Rich client user interface and utility components.
Server-side web client interface and reporting
components. Whatever type of Java development
you're doing, JClass can help.

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.



Evaluate and experience JClass today - visit:

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

ters interest in events by adding an event listener.

A final word about events: they are synchronous (*note*: don't confuse this term with the Java keyword "synchronized"). This means that your code will never be interrupted to handle an event. Events are only delivered when you call an SWT method. In the case of the event loop, calling `readAndDispatch()` allows any events that the user has generated to be delivered to your program. If you don't call `readAndDispatch()`, you'll never see events from the user; your program will appear hung, and the user will eventually kill it.

Listeners: Typed and Untyped

SWT has two ways to listen for events: typed and untyped.

A typed listener is exactly the same as a JavaBean listener. Typed listeners and their events are found in the package `org.eclipse.swt.events`. For example, to listen for a selection event in a push button, application code uses `addSelectionListener(SelectionListener listener)`. When the user selects the button, `widgetSelected(SelectionEvent event)` is called from the event loop. `SelectionListener` is an interface. If there's more than one method defined in the listener interface, an adapter class exists that provides default no-op implementations of the listener methods. This means that you could use a `SelectionAdapter` to determine when a button has been selected instead of a `SelectionListener`. Listing 1 adds two typed listeners to a button, listening for selection and a change in focus, respectively.

Untyped listeners provide a generic, low-level mechanism to listen for any event. There are only two classes involved: a single generic interface called `Listener` and a single event class called `Event`. These two classes are found in `org.eclipse.swt.widgets`. Instead of following the JavaBeans pattern that requires a specific method to add each kind of listener, untyped listeners are added using only one method, `addListener(int type, Listener listener)`. The type argument specifies the event you are interested in receiving. Type arguments are constants in the class `SWT` and are mixed case by convention. For example, `SWT.Selection` is an event type constant. All other constants in `SWT` are uppercase.

The `Listener` interface has a single method, `handleEvent(Event event)`, that's called when the event occurs.

One possible way to listen for untyped events is to

programmer and can be given back to the operating system. Controls are disposed using the `dispose()` method.

When the `dispose()` method is called, the underlying window system resources are released, giving memory and other operating system resources, such as handles, back to the operating system. When the root of a control hierarchy is disposed, the children are automatically disposed. Therefore, disposing a shell disposes the children. Similarly, disposing the display, disposes all the shells that were created on the display.

When a control is disposed, either explicitly from `dispose()` or implicitly when an ancestor is disposed, a `SWT.Dispose` event is sent. The dispose event is a good place to dispose graphics resources that you have created for that control. If you try to access a control that has been disposed, the operating system won't crash. Instead, SWT will raise an exception because it's a programming error to access a disposed control.

If you never dispose a control, but instead hide it or never make it visible, eventually the operating system will run out of resources. In practice, it's hard to write code that does this by accident. For one thing, programmers generally don't lose track of their controls because they require them to present information to the user. Because controls are usually visible, it's obvious when there are too many on the screen.

Positioning and Sizing Controls

Each control is sized and positioned relative to its parent using the methods `setLocation(int x, int y)`, `setSize(int width, int height)`, and `setBounds(int x, int y, int width, int height)`. The following code fragment positions and sizes three different controls:

```
control1.setLocation(12,23);
control2.setSize(100,100);
control3.setBounds(10,10,200,100);
```

Sizing and positioning of a control always refers to the entire control, not just the contents. For example, the size and position of a shell includes the window trim and the menu bar. The rectangle that contains the entire control is called the bounds of the control.

The client area of a control is the smaller rectangle within the bounds that the control uses to show its contents. For

“SWT PROVIDES LAYOUT CLASSES TO COMPUTE THE SIZE OF THE CONTROLS AND AUTOMATE POSITIONING OF CHILDREN WHEN THE PARENT IS RESIZED”

implement a single listener and use the event argument to determine the type of event that occurred (see Listing 2).

The trade-off between the two listener models is one of speed and space. Using untyped listeners, it's possible to minimize the number of classes and methods used to listen for events.

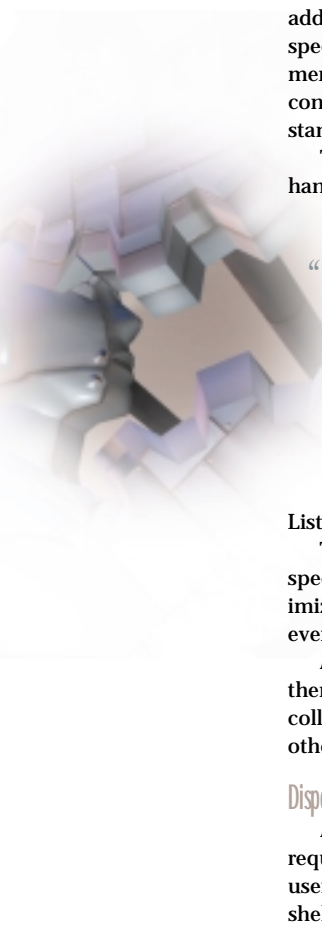
As well as adding listeners, it's also possible to remove them, but this is generally unnecessary. Listeners are garbage-collected when a control is disposed, providing there are no other references to the listener in the application program.

Disposing Controls

An SWT control is explicitly disposed when it's no longer required. Sometimes the dispose operation is initiated by the user. For example, the user may click on the close box of a shell. More often, a control is no longer required by the pro-

example, the client area of a text widget is the area where the lines of text are drawn, not including the scroll bars or borders. Child controls are always positioned relative to the client area of the parent, not its bounds. The client area of a control is

SWT PLATFORMS		
Operating System	-	Windowing subsystem
aix	-	motif
hpux	-	motif
linux	-	gtk
linux	-	motif
macos	-	carbon
qnx	-	photon
solaris	-	motif
win32	-	win32
win32-ce	-	win32



Remember when coding was easy? SOFIA does.

SOFIA™ is the Salmon Open Framework for Internet Applications, a new standard in Java development, built by **Salmon LLC**.

It is open-source code that you can download for free and put to work today.

SOFIA's powerful, scalable framework creates an approach to Java Web development that makes programming so easy even a ... well, you get the picture. **SOFIA:**



INTEGRATES TOOLS

■ SOFIA empowers J2EE programming with built-in integration of best-of-breed development tools—design tools like **Dreamweaver**, and IDEs like **IntelliJ** and **Eclipse**. SOFIA is anchored by J2EE class and tag libraries that offer more than 40 visual components (Button, Navigation Bar, DataTable) and 20 non-visual ones (DataStore, Internationalization, Task Scheduler).

SIMPLIFIES CODE

■ SOFIA's libraries provide more than **200,000 lines of ready-to-go code**. Based on the Model View Controller design pattern, SOFIA separates Java and HTML/JSP. The standardized approach means **less coding**, easier maintenance and debugging, and a fast learning curve. Instead of tedious, repetitive coding, programmers focus on cost- and time-efficient solutions.

IMPROVES PRODUCTIVITY

■ SOFIA enhances application development and leads to **real cost savings**. Its proven formula has evolved from more than **40 successful projects** led by Salmon LLC, a Java development firm with offices in New York, London and Sydney. This signature framework—battle-tested by clients that include Fortune 1000 companies—sets the new standard for Java development.

SOFIA SETS NEW BENCHMARKS

So, how does SOFIA measure up? We put the SOFIA .Pet to the test against the Sun J2EE Pet Store. The results:

	Sun J2EE Pet Store	SOFIA .Pet
Testing platform	J2EE reference implementation 1.3	Tomcat 4.03
Max throughput	6 pages per second (520,000 hits a day)	30 pages per second (2,600,000 hits a day)
Lines of code	22,221 lines in 250 files	3,563 lines in 50 files

Download SOFIA's open-source code today at salmonllc.com/sofia-jdj and discover why the other frameworks just don't measure up.





FIGURE 4 FillLayout with single control



FIGURE 5 FillLayout with 3 controls



FIGURE 6 RowLayout with 3 controls

never explicitly resized or positioned through API. Rather, setting the bounds of a control implicitly sets the bounds and client area of a text control. Figure 3 shows the bounds and client area of a text control.

Controls can compute their preferred size using the method `computeSize(int wHint, int hHint)`. This method doesn't actually set the size of a control but rather returns a size that's a good default for the control. When the constant `SWT.DEFAULT` is used for both the width and height hint arguments, the control computes the smallest size necessary to completely show its contents. When a

width or height value is used to compute the preferred size instead of the constant `SWT.DEFAULT`, the control computes its smallest size based on the hint arguments. For example, specifying a width of 100 and height of `SWT.DEFAULT` when computing the preferred size of a label that wraps is equivalent to asking the question: "If the label were to be resized to be 100 pixels wide, how tall would the label need to be to show the contents, wrapping lines of text as necessary?"

Use the `pack()` method of a control and set its bounds to the preferred size. Using `pack()` is equivalent to computing the preferred size and then using the result to set the size of the control.

As we've seen, the preferred size of a control depends on its contents. The preferred size can be queried from a control and used to position it within its parent. Writing code to explicitly size and position each control can be quite tedious and error prone. In addition, this code must almost always run when

done using the method `setLayoutData(Object layoutData)` on each control. Layout data and its API are very specific to the layout used by the composite. For maximum flexibility, layout data can be of any type but must match the type that is expected by the layout class. By convention, layout data classes end with the suffix "Data".

It's important to note that layouts are set on the parent while layout data is set on the children.

Two of the simplest layouts are `FillLayout` and `RowLayout`. More advanced positioning can be achieved using `GridLayout` and `FormLayout`. These last two layouts are more complicated but offer increased flexibility. The following section provides a quick overview of these four layout algorithms. In some situations it may be necessary to provide a customized layout algorithm, which can be achieved by subclassing the abstract class `org.eclipse.swt.widgets.Layout`.

FillLayout

`FillLayout` is intended to position a single child so that the child fills up all the available client area of a composite. `FillLayout` has no corresponding layout data. Figure 4 shows a shell using `FillLayout` with a single text control. The text control occupies all the available space in the shell. Currently, `FillLayout` provides no margins or spacing between controls.

If more than one control is added, the space is divided evenly between each of the controls, either vertically or horizontally. Figure 5 shows a horizontal `FillLayout` in a shell with three controls where the size of each is the height of the shell's client area and each control takes up one-third of the width.

In practice, you typically don't add more than one child to a `FillLayout` because it's not really intended to be used this way. Instead, you might use a `RowLayout`.

RowLayout

`RowLayout` lays out controls in a single row, either horizontally or vertically. Details such as wrapping of controls, margins, and the spacing between controls can be configured.

`RowLayout` has a corresponding layout data that's rarely

“ONE OF THE DESIGN GOALS OF SWT WAS TO CREATE A UI TOOLKIT THAT HAS A HIGH LEVEL OF INTEGRATION WITH THE OPERATING SYSTEM”

the parent is resized, so that the children will remain visible as the user resizes the window. SWT provides layout classes to compute the size of the controls and automate positioning of children when the parent is resized.

Layouts and Layout Data

Layouts are found in the package `org.eclipse.swt.layout`. Layouts are used to encode a positioning algorithm that's applied to the children of a composite whenever it's resized. The method `setLayout(Layout layout)` is used to set a layout into a composite. The details of the algorithm and API used to configure a layout are specific to each individual layout. Layout strategies are usually quite different and are built around different concepts, causing each layout to have a unique API. By convention, layout classes end with the suffix "Layout".

Some layouts allow each child to supply data that's specific to the positioning of the child, in addition to the preferred size, giving finer control over positioning and sizing. This is

used, called `RowData`. Normally, `RowLayout` uses `SWT.DEFAULT` when calling `computeSize()` to get the preferred size of a control. This happens when no `RowData` is provided for the control. When `RowData` is provided, it's used to specify values in place of `SWT.DEFAULT`, allowing you to explicitly set the size of a control within a `RowLayout`.

Unlike `FillLayout`, `RowLayout` wraps controls by default. Figure 6 show a `RowLayout` that has three controls on it. Although the label and button fit on the same line, the progress bar is placed beneath them because the shell has been resized so there's not enough space to show all three controls on the same row.

`RowLayout` is more flexible than `FillLayout` and is generally used to lay out rows of buttons. In practice, most windows are a lot more complicated and require more flexible layout algorithms such as `GridLayout`.

GridLayout

`GridLayout` divides a composite into a grid of rectangular

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

Object Reporting

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

Object Reporting means:

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



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

Easy Integration

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



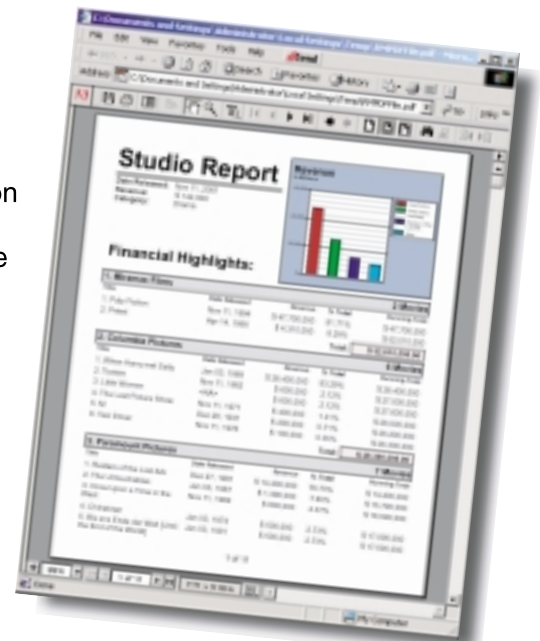
Fetch Objects and generate report in PDF and Flash

Dynamic PDF & Flash Output

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

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

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



View in browser, email or write to file!

Supports ColdFusion, WebLogic, WebObjects, WebSphere and any Java based application.

Visit our website to discover how easy and powerful Object Reporting can be!

Fully functional evaluation version available at <http://www.reportmill.com/webstart>

For more information, call (214) 513-1636 or send email to info@reportmill.com.



ReportMill

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

OBTAINING SWT

SWT was created as part of the IBM-led Eclipse open-source project whose members include Borland, Rational Software, RedHat, and TogetherSoft. Eclipse is an IDE that's implemented using SWT and is available at www.eclipse.org. On each platform, native widgets are used wherever possible. When a widget is not available, SWT provides an API-compatible emulated control.

As well as the `swt.jar` that contains the Java code required to run SWT, there's also a shared library that allows SWT to make operating system calls. The name of the library varies between operating systems, and on Windows is named `swt-xxx-nnnn.dll`, where `xxx` is the operating system and `nnnn` is a version number. When you use the Java command to run a program that uses SWT, you need to specify the location of the shared library. Assuming that you've just installed the Eclipse 2.0.1 to run a class `HelloWorld` contained in `helloworld.jar` on Windows, the Java command would be:

```
-classpath"helloworld.jar";C:\ECLIPSE\eclipse\plugins\org.eclipse.swt.win32_2.0.1\swt.jar
Djava.library.path=C:\ECLIPSE\eclipse\plugins\org.eclipse.swt.win32_2.0.1\os\win32\x86
HelloWorld
```

The directory used in the VM argument is the one that contains the file `swt-win32-2049.dll`, although the exact name and location of the library varies between Eclipse versions and operating systems.

cells. The number of columns is critical and determines the final number of rows. This depends on the number of children in the composite. By default, controls are placed into cells in the order they are created. As each child control is created, it's placed in the next available column, which, depending on the number of columns and the span of the control, may be on a new row. By default each control occupies one column.

GridLayout has corresponding layout data, called `GridData`, that's almost always used by the programmer. Using a `GridData`, the programmer can control the number of columns to be spanned by a control, the anchoring, the alignment against the edges of the cell, and the width and height values to be passed to `computeSize()`.

Figure 7 shows a `GridLayout` shell with two columns and three children that are two group boxes and a scale. The group boxes occupy the first and second columns, while the scale occupies both columns on the next line. The scale uses a `GridData` with a horizontal spanning of two, so the cell it will occupy will include both columns. To force the scale to fill the entire area of the cell, a horizontal fill property was also specified.

The "Gender" group box uses `GridLayout` with one column to place its two radio button children on separate lines. The "Likes" group box is using a `GridLayout` with two equal width columns. Its first two check buttons are placed in columns one and two on the first row, and the next two are placed on the second row. Because `GridLayout`'s "makeColumnsEqualWidth" property is true, this makes

"Music" and "Art" have the same width as the two wider buttons.

Using `GridLayout`, it's possible to construct sophisticated user interfaces and achieve just about any positioning and sizing requirement. Many user interfaces, especially those found in dialogs, are inherently grid-based, making `GridLayout` a good choice. For user interfaces that are not grid-based, `FormLayout` can be used.

FormLayout

A `FormLayout` allows you to specify the position of a control in terms of its edges. Edges of a control are attached to a position in the parent. This can be an absolute position, a fraction, or a percentage of the width or height of the parent. Edges can also be attached to the edge of another control, giving `FormLayout` maximum flexibility.

The corresponding layout data class `FormData` is almost always used by the programmer. Each `FormData` has fields that represent the top, left, right, and bottom edges of the control. Each field is an instance of a `FormAttachment`, which specifies the attachment to use for that edge.

The constructor `FormAttachment(int numerator, int denominator, int offset)` creates a new form attachment that's used to attach the edge of a control to a position in its parent. In Figure 8, the right edge of the list box is assigned the attachment `new FormAttachment(4,5,3)`, attaching it to be 4/5 of the width of the parent plus an extra 3 pixels. Instead of fractions, percentages can be specified.

The constructor `FormAttachment(Control control, int offset)` creates a new form attachment that's used to attach the edge of a control to the edge of a sibling. In Figure 8, the combo box attaches its left edge to be 5 pixels from the right edge of the slider using the attachment `FormAttachment(slider,5)`. By default, the opposite edge of a sibling is attached, although an alignment value can be provided to allow attachment to another edge. Under rare circumstances, it might make sense to attach the left edge of a control to the center or left edge of a sibling, potentially causing one control to be positioned on top of another.

Figure 8 shows a shell at its preferred size, while Figure 9

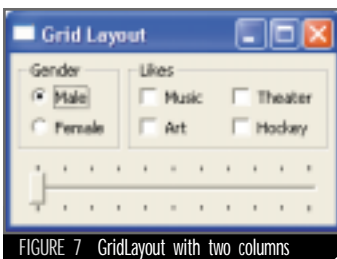


FIGURE 7 GridLayout with two columns

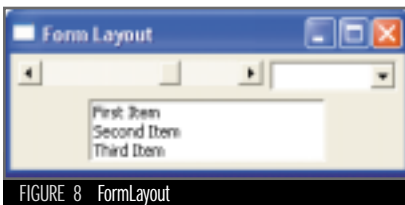


FIGURE 8 FormLayout

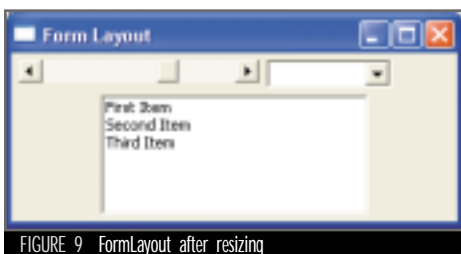


FIGURE 9 FormLayout after resizing

Listing 1

```
Button b = new Button(shell,SWT.PUSH);
b.addSelectionListener(
    new SelectionAdapter(){
        public void widgetSelected(
            SelectionEvent e){
            // button was selected
        }
    });
b.addFocusListener(
    new FocusAdapter(){
        public void focusGained(
            FocusEvent e){
            // button got focus
        }
    });
```

Listing 2

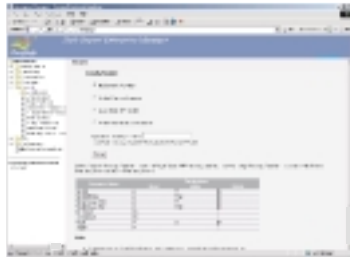
```
Button b = new Button(shell,SWT.PUSH);
Listener listener = new Listener() {
    public void handleEvent(Event event) {
        switch (event.type) {
            case SWT.Selection:
                // button was selected
                break;
            case SWT.FocusIn:
                // button got focus
                break;
        }
    }
};
b.addListener(SWT.Selection,listener);
b.addListener(SWT.FocusIn,listener);
```



Style Report

Information Vantage Point

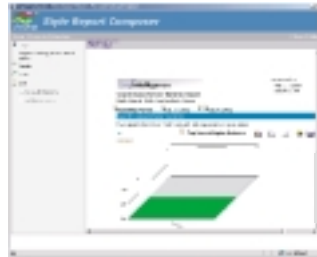
Java & Web Reporting Designed For Integration



Enterprise Manager

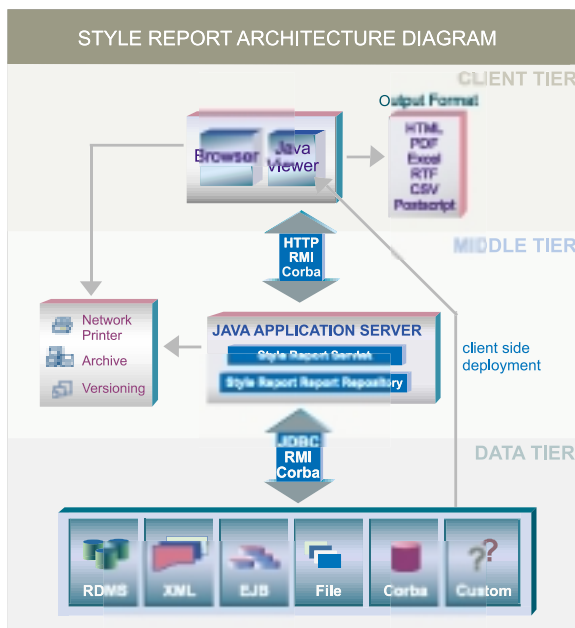


Report Designer



Web Report Composer

Style Report/EE blends the best of Java, Web and Enterprise Reporting to provide true Business Execution Intelligence. Based upon our award winning Style Report engine, the uncompromising One Report, Multiple Presentation capability allows you to concentrate on business logics. Through seamless integration with your applications, Style Report presents and delivers information when and where it's needed.



Style Report Features

Flexible Visual Designer	<input checked="" type="checkbox"/>
Flow/HTML layout	<input checked="" type="checkbox"/>
PDF, Excel, Word, CSV and other export format	<input checked="" type="checkbox"/>
Database/XML and other datasources	<input checked="" type="checkbox"/>
Zero Client Browser Viewing	<input checked="" type="checkbox"/>
Built-in Drilldown/Parameters	<input checked="" type="checkbox"/>
Zero Client Ad-hoc Reports	<input checked="" type="checkbox"/>
Virtual Private Security Model	<input checked="" type="checkbox"/>
J2EE drop-in deployment	<input checked="" type="checkbox"/>
Advanced Scheduling	<input checked="" type="checkbox"/>
Report versioning/Archive	<input checked="" type="checkbox"/>
Demand Paging/Report Cache	<input checked="" type="checkbox"/>
Portal/JSP integration	<input checked="" type="checkbox"/>
Full Web Service/Java API	<input checked="" type="checkbox"/>

 **InetSoft Technology**
open standards innovation

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

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

shows the same window resized. As the shell resizes, the controls are repositioned appropriately. The size of the list is automatically increased because of its attachments. The left, bottom, and right edges were expressed in terms of the parent, and the size of the parent has been increased. The slider and combo remain at their original positions. The combo, although attached to the slider, remains at the same location because the scale did not move.

As you can imagine, `FormLayout` is probably the most flexible layout of all. While it's possible to get almost any positioning imaginable using `FormLayout`, some user interfaces are more naturally expressed using the other layouts. Most SWT user interfaces are built from a combination of forms and grid layouts.

Forcing a Layout


When a composite is resized, its layout will automatically position and size the children based on the new size of the composite. There are times when the information that was used to calculate the position or size of a control can change without the layout being informed. For example, changing the text of a label will alter its preferred size. When this happens, the positioning and sizing calculations of the layout are no longer valid and children need to be repositioned even though the size of the parent has not changed. To force a layout to recalculate the position and size of its controls without resizing the composite, the `layout()` method is used.

Conclusion

This article has introduced some of the basic principles behind SWT and shows how to create simple windows and controls. One of the design goals of SWT was to create a UI toolkit that has a high level of integration with the operating

system. This is achieved by using native widgets and operating system resources. SWT has been ported to a number of different operating systems, giving it wide coverage on the desktop. Part 2 of this article will show how to use the graphics capabilities of SWT, as well as menus, tab folders, trees, and tables.

Resources

- *Eclipse project home page allows you to download Eclipse, including `swt.jar`, `swtsrc.zip`, and the SWT shared library:* www.eclipse.org
- *The SWT home page contains lots of code samples and on-line documentation:* <http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/platform-swt-home/dev.html>
- *Contains articles describing SWT, written by members of the SWT team:* www.eclipse.org/articles/index.html
- *PDF documentation for the Eclipse project including SWT:* www.eclipse.org/documentation/pdf/org.eclipse.platform.doc.isv.pdf
- *Contains useful links for SWT including how to run it on a Pocket PC, a list of frequently asked questions, and how to use SWT with native Active-X integration:* <http://eclipsewiki.swiki.net/2>
- *Gallery that shows many examples of SWT applications running on different operating systems and desktops:* <http://gallery.livemedia.com.au/> 

AUTHOR BIOS

Steve Northover is the principal architect of SWT. He is the SWT team lead for the Eclipse project and works at the IBM OTI Lab in Ottawa.

Joe Winchester is a developer for IBM in Hursley, UK, where he works on GUI software tooling for WebSphere.



J2ME



J2SE



J2EE



Home



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 the right 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>

New Enterprise Accounts on Dual Xeon Servers with 2GB RAM and RAID

Features	Basic	Professional	Enterprise	Dedicated
Monthly Cost	\$9	\$17	\$39	\$191
Servlet 2.3	✓	✓	✓	✓
JSP 1.2	✓	✓	✓	✓
EJB 1.1	✓	✓	✓	✓
EJB 2.x			✓	✓
Private JVM	✓	✓	✓	✓
Database	1	1	5	Unlimited
e-Commerce		✓	✓	✓
Tier 1 Center	✓	✓	✓	✓
Accounts/Server	100	100	25	1
Diskspace	30 MB	200 MB	900 MB	40 GB
Bandwidth/Month	3 GB	10 GB	15 GB	35 GB
Dedicated IP			✓	✓
Domains	1	2	5	Unlimited
Email Aliases	2	20	100	Unlimited
Servlet Contexts	1	1	10	Unlimited
Control Panel	✓	✓	✓	✓
FTP Accounts	1	1	5	Unlimited
Telnet/SSH	0	1	5	Unlimited
Web Mail	✓	✓	✓	✓
Web Stats		✓	✓	✓
Perl/PHP		✓	✓	✓
Failover			\$27/month	\$25/month
Engine Choices	✓	✓	✓	✓
WAR/EAR	✓	✓	✓	✓
Tomcat	3.x-4.0.3	3.x-4.0.3	3.x-4.1.x	3.x-4.1.x
JBoss	2.4.1	2.4.1	-3.x	-3.x
Cocoon	1.8.x	2.x	2.x	2.x
Backup	✓	✓	✓	✓



GLEN CORDREY J2ME EDITOR

Hello Everyone

Hello everyone. I'm Glen, the new kid at the helm of the J2ME section. I'll start by thanking Jason and Alan for the opportunity to expostulate in these hallowed pages. Like many Java developers I find **J2ME** an indispensable resource in my everyday work: its code and technical enlightenment have often saved me time and work, so I'm pleased to be associated with the magazine.

I'm particularly gratified to be involved with the J2ME section, because J2ME has been the most interesting and exciting technology I've worked with in awhile. I still work frequently with and enjoy J2EE, but J2EE has been around long enough to acquire a middle-age maturity, polish, and sophistication, while J2ME has the brashness and enthusiasm of a youngster. It's especially interesting to be part of the early years of a technology, when you can see its promise, grapple with some of the fundamental issues, and see that technology unfold.

And unfold it does. In January the JSR-185 expert group released the first official version of the Java Technology for the Wireless Industry (JTWI) roadmap, which lays out a time line for how the Java mobile phone platform will evolve as additional JSRs are released. One concern in the J2ME scene has been the possible fragmentation of the mobile phone platform as the result of OEMs adding proprietary APIs to provide functionality, such as HTTPS and multimedia, that was not specified in the MIDP 1.0 and related J2ME specifications. Much of this functionality is now being addressed by various JSRs (MIDP 2.0 mandates HTTPS, JSR-135 defines mobile media APIs), and the roadmap provides a near-term time line for the availability of these specifications so that developers and manufacturers can better plan their activities. It also promotes an "architecturally coherent" (to quote from the JSR) platform, which I can't believe anyone would think is a bad thing.

However, there's one thing about the JSR that annoys me, and so provides the opportunity for my first rant in these pages. The JSR contains yet another tongue-twisting acronym, JTWI. In his editorial last month (Vol. 8, issue 3), Jason talked about the "lack of brand awareness among the nontechnical public" re J2ME, and I think Sun's fondness for acronyms and the inability to come up with a user-friendly moniker for this technology is evidence of a critical blind spot. Sun has an engineering mentality and develops great technologies, but doesn't, from my viewpoint, appear all that strong in marketing.

Now engineering requires rational and analytical thinking, so we engineers (and apparently Sun) would like to think that if you simply build a better mousetrap, the world will recognize it as such and beat a path to your door. Unfortunately, the world doesn't work like that. Many people don't think analytically - if they did, do you think they'd be phoning psychics for life guidance? The better technology, and the better company, doesn't always win.

I think that the big stakeholders in J2ME - not just Sun, but also the handset and PDA manufacturers and other critical partners - need to come up with a better marketing approach for J2ME. Start with a catchy name and, if at all possible, some appealing graphic image that everyday folks can recognize and associate with the technology, even if they don't understand the technology itself.

As an example, look at Wi-Fi. A few years ago whenever I'd discuss 802.11b I was afraid the person I was talking to would get bored and walk away before I finished saying all those syllables. Now everybody says Wi-Fi, and nontechnies ask for Wi-Fi at the local consumer electronics-R-us store. They may not know what it is exactly, but they know it's what they need to wireless-enable their home or small business network.

Hello Everyone

I'm the new kid at the helm of the J2ME section. I'm particularly gratified to be involved with this section, because J2ME has been the most interesting and exciting technology I've worked with in awhile. I still work frequently with and enjoy J2EE, but

J2EE has been around long enough to acquire a middle-age maturity, polish, and sophistication, while J2ME has the brashness and enthusiasm of a

youngster.

by Glen Cordrey

50

J2ME: A European Point of View

Who are the key players in the European wireless application market and how does J2ME fit into this picture? In this article I look at the current trends in Europe regarding J2ME and how the more interesting Java Specifications Requests could possibly affect the European and global markets.

by Sven Haiges

52

Wi-Fi can also serve as a model from another angle. The Wi-Fi Alliance (www.wi-fi.net), comprised of 207 companies, has been instrumental in the success of Wi-Fi through their standards, certification, and marketing efforts. A similar effort on behalf of J2ME could help ensure the market success of the technology.

That's it for my first time on the soapbox. Thanks for your time, and so long until next month. ☛

glencordrey@sys-con.com

AUTHOR BIO

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.

*IT'S NOT JUST
A CONFERENCE.
IT'S A CHANCE
TO CONNECT
WITH PEOPLE.*



*LIKE THE
MILLIONS
ALREADY
USING
BREW™*



Nothing connects you with the wireless world faster than the BREW 2003 Developers Conference. It's the easiest way for you to establish connections with the world's leading wireless carriers and millions of their subscribers. Take advantage of technical tracks, guest speakers, business sessions and more. Make the connection. Register at www.brew2003.com.



**BREW 2003
DEVELOPERS
CONFERENCE**

**Register Now:
www.brew2003.com**


brew™
2003
April 28 - 30
San Diego, California



J2ME: A European Point of View

written by Sven Haiges

A wireless wonderland for Java

Who are the key players in the European wireless application market and how does J2ME fit into this picture? In this article I look at the current trends in Europe regarding J2ME and how the more interesting Java Specifications Requests could possibly affect the European and global markets.

Europe: One Standard=Fast Growth

By the end of 2002, there were a total of 365 million mobile phone subscriptions in Europe. According to the researcher EMC, this number will grow to 413 million subscribers in 2004 – a mobile phone penetration of 83%. At the end of 2001, more than 50 of Europe's 76 mobile operators had already launched their 2.5G (GPRS) networks and currently GPRS is available almost everywhere. Austria launched a 3G (UMTS) network in the fall of 2002 and Europe's largest mobile operator – Vodafone with over 100 million subscribers – will launch its UMTS network this summer in Germany. The fast growth in earlier times was due to the commitment to a single communication standard: GSM. While the U.S. still struggles with building a nationwide GSM network to switch to 2.5G GPRS and then move to UMTS (W-CDMA), most countries in Europe will have their UMTS networks up and running by the end of 2003.

Shift Toward Data Communication

Every mobile operator will tell you that there's a shift toward data communication, and it's unclear if this shift is a major opportunity for mobile operators or a major threat. The business models and growth projections of many European mobile operators are built upon the assumption that data communication will increase sharply with the introduction of the packet-based 2.5G and 3G networks. While the introduction of 2.5G networks has increased the usage of the Wireless Application Protocol (WAP), we're still waiting for bigger growth. The biggest European success story for data communication is still the Short Message Service (SMS): Europeans and especially young Europeans are crazy about sending over-priced 160 character messages (about 0.15 euros each) to each other. At the beginning of 2002, Siemens shipped its first mobile phones with "battle games" that use SMS to communicate with peers.

The latest trend in Europe is picture and video messaging. Vodafone, Europe's largest mobile operator, was the first to

introduce a mobile phone with a built-in camera, Nokia's 7560. This quite heavy and large phone soon had competition from smaller phones made by Sharp and Panasonic that Vodafone is selling under the Vodafone Life! label. Vodafone succeeded in integrating its own look and feel into the mobile phone – they've completely branded the phone menu.

How Does J2ME Fit In?

You don't have to search that long to find a Java-enabled mobile phone in Europe. You almost can't buy a phone without it any more! When it comes to the penetration of Java-enabled phones, Europe is the second wonderland (probably after Japan). The strong brands like Nokia and Siemens are committed to the Java platform and almost any device ships with support for MIDP 1.0 (based on the CLDC 1.0).

The first Java-enabled devices were introduced in the summer of 2001: the SL45 was a much-hyped phone that provided Siemens with heightened brand awareness. Since then, several bugs have been fixed and the currently available phones are finally quite stable and predictable. Also note that the SL45 first shipped without support for GPRS, which meant you had to wait up to one minute to establish a data connection, if it could be established at all.

GPRS means always-on functionality and immediate access to data networks. This is different with GSM, where you first have to establish a connection, then the phone has to register with the network, and finally a transfer can begin. It took a long time to establish a data connection with the SL45i, which reminded most of the early adopters of their experiences with the first phones that implemented WAP – wait until the phone finally tells you that it can't establish a connection!

Although many phones are currently Java-enabled, this doesn't mean that the paradigm "write once, run everywhere" is true for the J2ME world. Each manufacturer has added some supplemental APIs to the core J2ME APIs (that are based on the APIs of CLDC and MIDP) that make it hard to produce cross phone-compatible software products. It's possible in theory, but the additional features that these APIs provide are hard to ignore as a developer: the sending and receiving of SMS messages, special gaming APIs, and control of the back light or the vibrating function are among those features that you want to use if you're developing "cool" Java-based games. Fortunately, most of those features are addressed by Java Specification Requests developed through the Java Community Process.

In addition, Sun is addressing the increasing API fragmentation with JSR 185, which provides a Java technology roadmap. The first roadmap is available at the JCP Web site,



and new roadmaps are expected to be issued every six to nine months. In short, the first roadmap says that Java-enabled mobile phones that will be available in Q2/Q3 must have the following characteristics to be compliant with this roadmap: support for CLDC 1.0 or CLDC 1.1, Wireless Messaging API 1.0, and MIDP 2.0. Further, the Multimedia Messaging API (MMAPI) is conditionally required if the device exposes video playback, or audio or video/image recording capabilities to Java applications.

J2ME-Based Applications and Billing Options

When it comes to currently installed and used J2ME applications in Europe, the predominant application type is clearly mobile games. The available Java-enabled handsets have created both a new demand for mobile games and lots of new Web sites that sell mobile games. Java gaming revenues are estimated to increase to \$195 million by 2004 (Source: O2, Germany).

To install the applications, there are basically two options to choose from: download the software and transfer it using a data link cable or directly download it over-the-air (OTA) using the mobile phone's WAP browser. Because most mobile phone users do not use their WAP browsers on a regular basis and thus would never visit the vendor's WAP site to install an application, and the

currently try to dominate as many parts of the wireless value chain as possible and therefore try to sell their own games over their own channels. They recognize that there's a huge number of software developers who can provide them with new games.

Some mobile handset manufacturers are giving away their games to provide additional value to their customers. Siemens, for example, provides free Java downloads via their wap.my-siemens.com/games site that you can access with a WAP-enabled mobile phone.

Regarding the billing of data transfers, most operators currently bill the kilobytes transferred over the network, as most networks are GPRS-enabled and most cellular phones come shipped ready to use GPRS. Also, GPRS is always available; you don't have to buy a special Internet plan or pay extra to have it activated. The prices are quite high and are related to the usage of the Wireless Application Protocol or the downloading of mobile Java applications, which does not produce high-volume transfers. If you apply these data plans and try to download, say a 4MB MP3 file, you'd pay about \$217! Note that the current prices for GPRS transfers should only be considered when browsing WAP sites or downloading Java applications (about 50KB per application).

There are also some volume plans that you can purchase that offer you better prices: 20MB for about \$42. This would

“Although many phones are currently Java-enabled, this doesn't mean that the paradigm ‘write once, run everywhere’ is true for the J2ME world”

installation using a data link cable is too complicated, some vendors send their customers SMS messages that allow them to connect directly to the WAP site for the application download. This interesting solution also solves the payment problem:

1. After the user has surfed the mobile games vendor's Web site and decided which game to download, he first has to call a premium-charged telephone number to obtain a transaction number (TAN). This service can easily be set up using the VoiceXML language that allows the rapid setup of voice-enabled applications.
2. When the user then wants to buy the chosen game, he also has to enter his mobile phone number and the TAN number. The TAN number is used to verify that the user has paid and is valid only once. The mobile phone number is used to send the user a special text message that will allow him to download the game.
3. The user immediately receives a text message. The message contains a special URL that points to the Java game's installation files. By clicking on the phone's call button, most phones directly start the WAP browser and begin requesting this URL. This will result in the download (OTA) of the J2ME application. After the installation, the user can access the game through the phone's menu.

Sounds complicated? It definitely is. Another solution application vendors use is a voice-enabled service only: the user first notes the order ID of the application she wants to download, and then calls the service telephone number. She must then enter this order ID and her telephone number to receive the text message with the download URL and then proceeds as described above.

Most third-party software providers have to use these methods because the mobile operators won't let them use their own billing services, or the billing services are not yet capable of dealing with Java application downloads. Most European operators

allow you to check your e-mails or surf some Web sites for a fair cost, but it still can't be considered cheap or mainstream. In reality, few people use GPRS to surf the Web. Some people use their phones to access WAP sites to get some news or to buy movie tickets. The German operator O2 has a very easy billing solution: each click on a WAP page costs .05 euro (5.4 U.S. cents). This is a billing solution that the users understand – when accessing WAP sites you normally don't see a data counter, so users are always unsure of the costs.

The prices will drop significantly when more bandwidth becomes available with UMTS, especially for the mobile Internet. Operators announced that there would be no special fee to use UMTS for voice communication, only new features like video streaming will be extra.

J2ME Java Specification Requests

I'll now look at current Java Specification Requests and their impact on the European market. Table 1 shows you the JSRs that I picked from the 41 currently available at jcp.org and will discuss in this article.

JSR 120	Wireless Messaging API
JSR 205	Wireless Messaging API 2.0
JSR 135	Mobile Media API
JSR 179	Location API for J2ME

Table 1: Most interesting Java Specification Requests

Wireless Messaging API

Under the lead of Jan Eichholz from Siemens AG in Munich, the Wireless Messaging API (JSR 120) was first released in August 2002 and allows access to the SMS capabilities of the mobile phone. Further, Cell Broadcasts (messages broadcast over the mobile operator's network to every mobile phone) can be accessed. An enhancement to this API that also covers the

DARE TO COMPARE?



RIP OUT THIS FEATURE LIST AND SEE HOW YOUR SERVER STACKS UP AGAINST MACROMEDIA® JRUN™ 4 SERVER

NEW AND ENHANCED FEATURES OF JRUN 4:	JRUN 4	
Certified J2EE 1.3 Compatible	✓	
High performance web container with JSP 1.2 and servlet 2.3 support	✓	
Advanced EJB 2.0 support including local interfaces, message-driven beans, EJB-QL, and CMP 2.0	✓	
Additional Java® 2 APIs -- JMS 1.0.2, JTA 1.0, JAAS 1.0, JCA 1.0, JAXP 1.1, JavaMail 1.1, JDBC 2.0, JNDI 1.2, RMI/IIOP 1.0	✓	
Support for XML, SOAP, WSDL, and UDDI for web services	✓	
Built-in production quality HTTP server	✓	
Hot modification and auto deployment of applications	✓	
Integrated XDoclet support for simplified component deployment	✓	
Built-in Macromedia® Flash® MX Remoting services for high impact user interfaces	✓	
Type IV JDBC drivers included for Oracle®, MS SQL Server, IBM® DB2, Sybase®, and Informix®	✓	
Highly customizable and extensible JMX-based service-based architecture	✓	
JMX-enabled web-based management console	✓	
Advanced multi-tier server clustering enabled by Jini® technology.	✓	
Leading OS platforms, including Unix, Linux®, and Windows®	✓	
FULL-FEATURED PRODUCTION DEPLOYMENT	\$899/CPU	



**Java Compatible™
Enterprise Edition**

We've packed JRun 4 with features to make it faster, easier, or more affordable than any other J2EE server you can find. Don't just trust what you read, try it out yourself. You'll see why over 10,000 leading organizations deploy their J2EE applications on JRun.

Our **FREE** non-expiring developer version will get you started. Download JRun 4 today for **FREE** at:
www.macromedia.com/go/jrun4jdj

Go ahead...we dare you.

Copyright 2002 Macromedia, Inc. All rights reserved. Macromedia, JRun, and the JRun logo are trademarks or registered trademarks of Macromedia, Inc., which may be registered in the United States and other countries. Jini, Java and Java-based trademarks and logos are the trademarks or registered trademarks of Sun Microsystems Inc., in the United States and other countries. IBM and Informix are registered trademarks of IBM Corporation. Sybase is a registered trademark of Sybase, Inc. Linux is a registered trademark of Linus Torvalds. Windows is a registered trademark of Microsoft Corporation. Oracle is a registered trademark of Oracle Corporation. All other trademarks are property of their respective owners.



**macromedia®
JRUN™ 4**

current trend of picture messaging – MMS – is the Wireless Messaging API 2.0 (JSR 205) currently under development.

The advantage of this API is that it unifies all the proprietary solutions that each mobile phone manufacturer had implemented in their early Java-enabled phones. SMS is a simple, yet powerful way of communicating with peers or other systems as gaming servers. It's instantly available on most mobile phones and has predictable costs, which is crucial for the end user.

As the need for such messaging support is high in Europe and solutions are already developed and only have to be adapted to the new API, we'll start to see implementations of this API in the next few months. Implementations for version 2.0 with MMS support are expected to hit the shelves next year.

SMS is the ideal communication service for mobile gaming applications: its ubiquitous availability makes it easy for developers to build mobile games that harness this technical possibility and exchange information with other players (high scores, moves in a chess game, etc.). Further, each outgoing SMS is billed by the mobile operators, which makes them very willing to promote those games within their mobile ports. Such games could be offered for free and would earn revenues through the resulting text messaging traffic. The application developers would earn money through revenue-sharing agreements with the mobile operators.

Mobile Media API 1.0

This API (already released) was led by Jyri Huopaniemi (Nokia) and the first products should be available now. The

Problems may arise because this API could endanger some business models of mobile operators who expect to sell this location information to the users. Onboard location information, which is gathered through the use of a small GPS receiver within a mobile phone, is available for free and undermines the mobile operators plans. The operators favor a network-assisted solution, whereby the mobile network does the positioning and sends that information to the client device.

The regulatory office of Germany (RegTP) estimates that the total European market for location-based services will increase to \$9,167 million in 2005. For the end user, location-based services will solve questions like: "Where is the next restaurant?" or "Where is the next ATM?" So far, location-based services were developed only for WAP-based applications and needed strong support from each mobile operator. This API makes it possible to use the same implementation with several mobile operators.

Conclusion

J2ME offers many possibilities for both handset manufacturers and mobile operators, which may be the main reason for its success in Europe. As the market is almost saturated, mobile operators fight for customers and want to differentiate themselves through the services they offer.

Stateside, mobile phone users see their phones mostly as an extension of their telephone line, whereas in Europe it's much more than voice communication: right now it begins with sending text, picture, or video messages, and ends up with surfing the mobile Internet (WAP). The mobile phone is

"As major European providers launch their 3G networks this year and increased bandwidth becomes available on mobile devices, new services become feasible"

Nokia 3650 is one of the first cellular phones to implement this API. It allows for the use of the multimedia capabilities of the device, such as sound and video, through a generic interface and is extensible for future demands.

As major European providers launch their 3G networks this year and increased bandwidth becomes available on mobile devices, new services become feasible. These services include the on-demand usage of the latest TV news on a mobile device or the download of the latest MP3 to the phone's memory. The success of such services will depend on the pricing model for data transfer that the mobile operators will introduce. Data consumed by surfing mobile Web sites (WAP) should be treated differently than downloading a 4MB MP3 from a music shop. From the device manufacturer's point of view, playback of videos or streaming audio is no problem. Unfortunately, it will take some time until most manufacturers make these proprietary APIs accessible via Java.

Location API

Access to current location information is a crucial component for all future wireless applications. This information is highly valuable as it makes it possible to design smart applications that link the location of the user to their activities and more. Since February 23, this API has been available for public review at the JSP Web site. Under the lead of Kimmo Löytänä (Nokia), a common interface for accessing location information – both network assisted and information that's readily available on the device via GPS – has been designed.

The final release of the specification can be expected in the fall of 2003 or even later. First implementations will take even longer, as several hardware solutions have to be evaluated.

becoming an irreplaceable element of everyday life.

This is also a big opportunity for third-party software developers. As we switch from 2G-based networks to 3G, completely new links in the wireless value chain emerge. The need for content and applications is getting stronger as more and more users get the latest technology in their hands when replacing old phones.

To leverage the power of both the wireless Internet and your Java applications, several key players in the industry have to work together. Most of all, mobile operators have to provide billing solutions for third-party developers and have to forgo using walled-garden approaches.

Resources

- *Java Community Process*: www.jcp.org
- *EMC database*: www.emc-database.com
- *Mobile Messages (SMS, EMS, MMS) in Western Europe, Gartner Dataquest, 2002*: www.digitwireless.com/press/articles/gartner_excerpt-08nov01.pdf, www.intromobile.com/mobilemessaging/mobilemessaging_01_e.htm
- *Market for location-related services, Strategy Analytics, 2002*.
- *World market for Mobile Gaming, Frost & Sullivan, 08/2002*: www.mindbranch.com/catalog/product.jsp?code=R1-2233
- *Handspiele – mobile gaming*: Thomas Kern, O2 Deutschland Presentation while the conference Münchner Medientage in Munich. ☛

AUTHOR BIO

Sven Haiges is a student of computer science and business at the University of Furtwangen, Germany, and currently lives in Vancouver, Canada, where he is completing an MBA in management of technology at Simon Fraser University.

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



Take a tour of PerformaSure today:

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

Tifosi 2002 Enterprise Integrator

by Fiorano Software

REVIEWED BY LAURENT MICHALKOVIC laurent_m@hotmail.com

Fiorano Software Inc.
718 University Avenue, Suite 212 Los Gatos, CA 95032
Phone: 408 354-3210
E-mail: info@fiorano.com
Web: www.fiorano.com
Specifications
Platform: Any platform supporting JavaSoft/IBM/Symantec JVM 1.3 and above
Test Environment
Production: Two desktops, Windows 2000 Pro/Pentium 4-1.6GHz/512MB
Development: One laptop, Windows XP/Pentium III 1.0GHz/512MB

info

Java started as a simple programming language with a rather comprehensive runtime library. To penetrate the enterprise world, Sun then created the J2EE specification and, consequently, their application server.

Workflow management applications represent the next logical evolutionary step in enterprise development. They help model and implement standardized business processes by offering, out of the box, the advanced data integration and process automation functionalities required by many modern enterprise projects.

Network Computing

Capitalizing on its experience with Fiorano MQ (one of the first JMS implementations), Fiorano Software created Tifosi, a new breed of integration middleware with advanced workflow management capabilities that's based on a distributed computing model. The concept is simple: route XML messages between software components connected via input and output ports.

Unlike traditional workflow managers that use a monolithic architecture where all the code is executing on one single computer, Tifosi (see Figure 1) uses distributed Peer Servers (TPS) that act as containers for the business services (the individual tasks in the process). This network-centric architecture provides a cheap way to scale your application by simply adding extra peer servers.

Upon installation, the Tifosi program group contains individual icons for 3 servers, 7 administration tools, 12 PDF manuals, and 2 CRM/ERP demo applications. Both the Concepts and the Evaluation Guides are good starting points for any new Tifosi user. By default, the application comes with a single TPS. Adding more is a matter of running Setup again and selecting "Workstation Edition." To simplify administration, the Enterprise Server (the heart of Tifosi) automatically detects new TPS on the network. Alternatively, the administrator can use the Services and Security Manager to declare them manually. The list of functionalities covered by the administration tools is too long to discuss here, but Fiorano has included GUI-based applications to monitor the servers, workflow instances, documents, messages, message queues, services, permissions, etc.

Workflow Applications

Tifosi applications are created by visually laying out icons that represent individual services using the Distributed Application Composer (TDAC). Besides the default properties, some services can be further configured using a custom properties sheet (writing

these is your job when creating custom services). To link the services, you can choose between two types of routes:

- **Hub-based:** All messages transit on the JMS server (Tifosi includes Fiorano MQ, but the documentation states that you can replace the JMS engine).
- **Peer-to-peer:** Messages are sent directly from one TPS to the other.

This is an important decision motivated by your scalability, reliability, and performance needs. Experiment with both, as Tifosi allows you to switch between routes at the click of a button, even on a production system.

The XML format for saving the application is proprietary, but Fiorano told me that they are working on compliance with emerging standards to allow for the exchange of applications with other workflow managers.

Tifosi comes with a large number of pre-built services, organized by categories on a floating toolbar inside the TDAC. These services include:

- Web connectivity, e.g., sending or receiving XML documents via HTTP, and accessing a SOAP-based Web service.
- Bridging with MQSeries, EJBs, FTP, JMS, MSMQ, SMS, and SMTP
- Database adapters – the RDBMS adapter is very powerful and took care of all my database needs
- Transforming data, e.g., compressing, decompressing, encrypting, decrypting, or applying XSLT stylesheets
- Executing scripts written in JavaScript, Perl, and Python
- Other functions include writing files, basic flow control logic, openadaptor connectivity (see www.openadaptor.org), and monitoring disk space or files

Custom Services

Despite this long list, there are times when you'll need to include your own business logic. Fiorano provides two options. For the simpler cases, you can use one of the scripting services. Alternatively, you can use Java, C, C++, or any COM-aware language to create your own services. A C# library is available if you want to build a .NET client.

The service creation wizard (launch the Tifosi Services & Security Manager, then select Create New Service in the Actions menu) generates fully functional Java services with a default custom properties sheet. The source code is well commented and uses "//TODO" annotations where you should add your customizations. At the time of writing, Tifosi only offered to generate a project file for Oracle JDeveloper and Borland JBuilder. Considering its popularity, I hope support for Ant is next on Fiorano's features list.

Fiorano

Integration Middleware

Your one stop solution center for Enterprise Application Integration (EAI), BPM, Distributed Workflow and Messaging Middleware.

FioranoMQ JMS Server - Enterprise Messaging Infrastructure	Tifosi™ ESB Enterprise Service Bus Mid Tier Integration Solution	Tifosi Enterprise Integrator Comprehensive full featured Integration Broker Suite
Fastest, Most Scalable JMS Server	Web Services Integration including support for SOAP, WSDL, UDDI	100+ Packaged and Legacy Application Adapters
24x7 Reliability and Hot Failover	Service Oriented Architecture	EAI, BPM and Distributed Workflow
Standards based	High Availability	Comprehensive Security Management
JMS 1.1 compliant	Application Server Neutral	Multiple transports (JMS, MQSeries)
Supports Multi platform and languages	Works with other MOM infrastructure	Monitoring and Document Tracking
Bridges to MQSeries and other MOM Infrastructure	Easily integrates with J2EE, .NET, COM	Distributed Deployment and Configuration Management
Easy upgrade to Tifosi	Easy upgrade to Tifosi Integrator	Dynamic Logging and Versioning

Solutions to satisfy your technical and financial challenges
Implement in stages, with incremental Return on Investment

Fiorano is #1 in Standards based integration

Fiorano
Enabling Change at the Speed of Thought™

Download Free Evaluation Software today at www.fiorano.com

A Real-Life Scenario

A prominent travel agency that provides corporate travel services hired me to help improve the managing of the Customer Loyalty Program for a large Canadian bank. In identifying the project's success criteria, we identified the need to increase the agents' accountability for their involvement in the process, to provide traceability of each customer's operations, to access real-time status information, and to integrate in real time with third-party systems. To mitigate the risks, the agency wanted a rapid proof of concept that could be turned into the final system in a continuous iterative process, akin to changing a tire on a moving vehicle. Time-to-market was also essential, so I looked for a product capable of integrating with existing applications with minimum changes to the source code. The end-to-end system had to be operational even though not all the subcomponents would be ready at the same time. Vendor neutrality and support for COM clients were also imperative. The loyalty program was fairly new so maintainability and scalability were high priorities. The system would initially keep track of 2,500 new operations per week, some taking up to a month to complete, and grow with the popularity of the program. Last but not least, the agency could not afford to pay the typical Enterprise price tag. With these criteria in mind, Tifosi came out on top, offering an excellent compromise between cost, complexity, and features.

Using the free 45-day evaluation version (www.fiorano.com/tifosi) and with no prior exposure to the product, it took me two weeks to build the first workflow application (see Figure 2). Fiorano's customer service was instrumental in identifying some problems we had with the Tifosi ActiveX control, and in the process incorporated some suggestions we made. Working with Fiorano's customer service was very productive, and I never had to wait more than a day for an answer. The simple client-side APIs allow the programmer to focus on the business code rather than on plumbing details. To further simplify coding the custom services, we did not use a database (a common design in workflow applications), but included all the business data inside the XML messages. Despite the amount of data involved, 500 new messages (each between

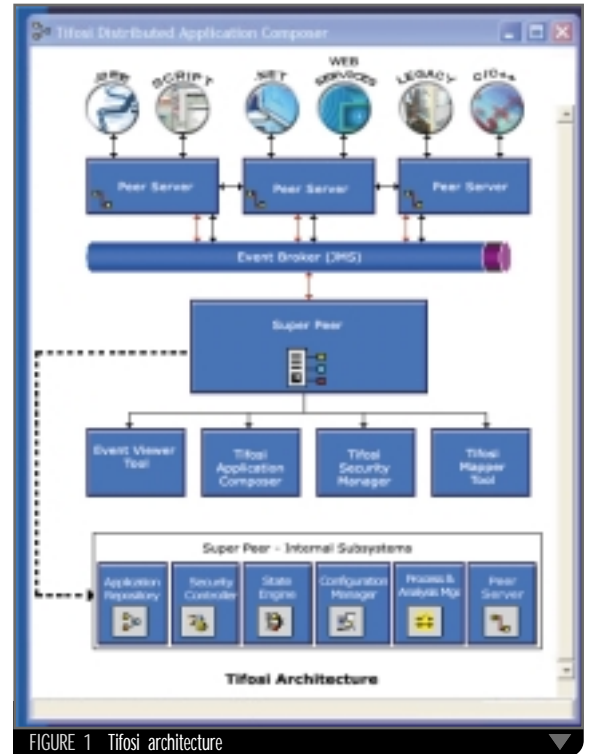


FIGURE 1 Tifosi architecture

100K and 200K) injected into the system daily, Tifosi performed flawlessly.

Final Word

The underlying concepts of Tifosi are clear and simple to understand. The documentation is straightforward and detailed. Furthermore, Fiorano has included many more functionalities than I covered in this review. These include the XML Mapper, a full-fledged GUI-based XSLT stylesheet generator, and the XML content routing engine (based on the results of XPath queries). I would definitely recommend Tifosi for business-process management or enterprise-integration problems similar to mine. My only serious request of Tifosi: some support for running the TPS as Windows NT services. ☛

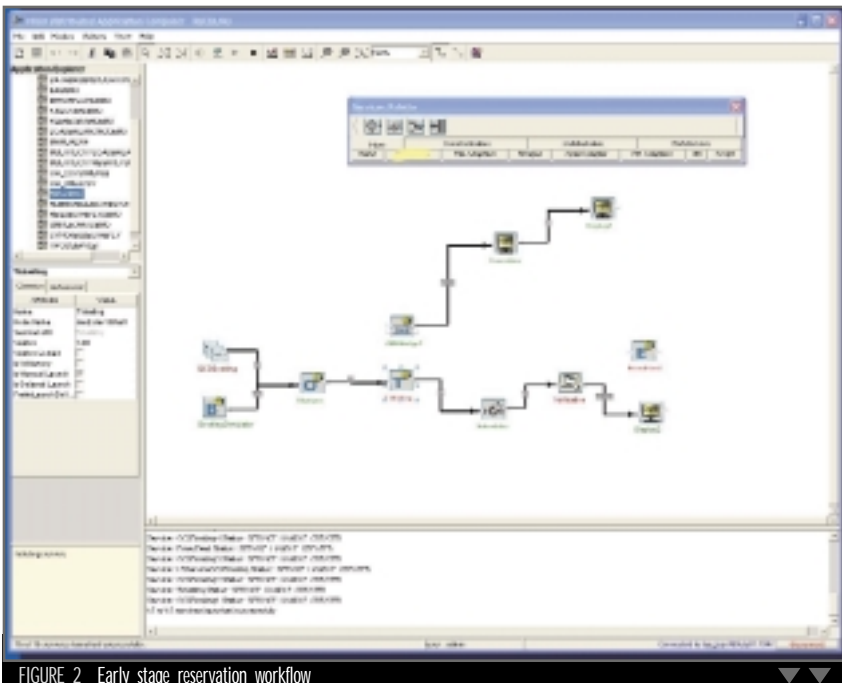


FIGURE 2 Early stage reservation workflow

Product Snapshot

Target Audience: Business analysts, technical architects, consultants, Java programmers

Level: Some experience with distributed messaging systems; good systems design skills

Pros:

- Solid integration middleware and workflow manager with many services and adapters
- Includes lots of example applications and source code for most of the built-in services
- Visual debugger to display the workflow messages in real-time
- All server applications are managed remotely

Con:

- No support for executing TPS as NT services on headless servers

Saturday Sessions

Alan Williamson
JDJ Editor-in-Chief

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!**



JDJ Workshop with Alan Williamson

M	T	W	T	F	S	S
---	---	---	---	---	---	---

Coming to you...

April: NEW YORK WASHINGTON, DC	May: BOSTON TORONTO	June: ATLANTA RALEIGH
---	----------------------------------	------------------------------------

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



To Register

www.sys-con.com/education

Call 201 802-3058

SPONSORED BY

JAVA DEVELOPER'S JOURNAL

PRODUCED BY
SYS-CON EVENTS

PRESENTED BY
i-TECHNOLOGY EDUCATION

Conquer-IT! J2EE

by Trans-World Resources, LLC

REVIEWED BY
TIMOTHY RONDEAU & DUNCAN MURPHY
timothy_rondeau@symantec.com
duncan_murphy_ii@symantec.com

info

- Trans-World Resources, LLC**
8 Periwinkle Circle
Tinton Falls, NJ 07712
Phone: 732 493-9449, 866 500-9449
Fax: 732 695-2770
E-mail: enquire@twresources.com
Web: www.twresources.com
- U.K.**
11 Brookdere Drive
Northwood Middlesex - HA6 3NS
Phone: +44-1923-822-998
- Test Environment**
Windows 2000 sp3, IBM ThinkPad
T30 Mobile Intel Pentium 4 - M 2GHz,
256MB

There are seemingly countless self-paced J2EE computer- and Web-based training tools on the market today, and many are of questionable value. However, Conquer-IT! J2EE Part 1: JSPs and Servlets stands out from the crowd, focusing on the key skills developers require and giving users a chance to work with an actual application server, real code, and associated build procedures – very different from standard multimedia simulators found in most computer-based training products.

From a content perspective, the Conquer-IT! CBT focuses on the critical skills developers need to be productive when they start their first programming assignment (the 20% of skills J2EE developers use 80% of the time). One interesting approach is the inclusion of a Best Practices section in which the developer includes recommendations based on actual field experience.

Product Description

Conquer-IT! uses several techniques for imparting J2EE knowledge, including animation, interactive code analysis, and lab samples with build scripts, allowing the student to deploy on Oracle9iAS, BEA WebLogic, or IBM WebSphere. The entire C/WBT is narrated, and a scripted manual is provided for users to follow the narrations.

Each section of Conquer-IT! begins with an introduction of concepts. In these sections, the narration explains the concepts and bullet points are displayed on the screen to reinforce the narration. This introductory section sets the stage for any examples that will be used to illustrate how to implement these concepts.

Following the concept screen(s), interactive coding analysis sections show the user how to write code to implement the concept by walking the user through the programming model using an example. Each line of code is highlighted, and a voice clearly explains the code and its effects.

The user is then encouraged to open a command window and run build scripts on included samples that tie back to the section in question. After testing the deployment, users are encouraged to review the solution to ensure that they understand the concepts (see Figure 1).

Finally, powerful animations illustrate conceptually what happens within an application server when the code is deployed. These animations bind the code and the concepts together, leaving a clear understanding etched in the user's mind (see Figure 2).

Using the Tool

Installation

Installing the CBT was a nominal task. The CD comes with an executable, a LabExercises

directory, and a PDF Documents directory. The only thing to watch out for during installation is that the default build scripts assume that the CBT has been installed to the C:\ directory. If you change the installation directory to be other than C:\, there are clear instructions in the read-along book that explain the necessary adjustments.

Usage

Learning through a CBT can sometimes be a challenge since CBTs don't offer the interaction that the classroom provides. This is why the ease of use of the CBT is extremely important in facilitating the learning process in the absence of an instructor. Using the Conquer-IT! tool is quite simple. All I had to do was run the executable, and it was off to the races. I found the interface easy to use; there are six buttons on the screen with all the basic CBT navigation functionality: Review, Replay, Next, Pause, Resume, and Main Menu. So it's pretty simple to get around efficiently. One thing that was a bit bothersome was the fact that the replay button takes you back to the beginning of the section and not just the current screen. It's my understanding that this has already been fixed in the next version.

Usually, the worst part of CBT is having to listen to the computer-generated voice for an extended period of time. The narration in this CBT is exceptional. The voice kept me interested, and more important, awake! This, paired with the visual effects and the read-along book, made for a good learning experience. One benefit of having the read-along book is that when you're tired of listening to the narration you can focus your attention on the book.

We all know how frustrating it is to be sitting in class, or through a CBT, and have either the instructor or the CBT cover a concept way too fast. One of the good things about the ConquerIT! CBT is that the coding sections and animations require interaction, which provided me with enough time to assimilate the concept prior to moving to the next steps. I never felt rushed through a section.

LabExercises

The lab exercises matched the interactive code analysis sections. They were useful to the point that I was able to look at an actual code sample. If I wanted to really learn anything outside of the covered code analysis sections, I'd have to experiment with the code samples and redeploy. I found this to be good and bad. Good in that there are real working samples that I could experiment with; bad because I wasn't forced to sit and follow lab instructions to build my own components. This is less a knock on the CBT and more of a problem of learning outside of the classroom environment.

I felt that while it's appropriate to use simple examples for illustrating the concepts, a few more could be added – it never hurts to have a lot of examples. What's provided is def-

Premiering
June 2003
at
JavaOne

www.sys-con.com



Millions of Linux Users One Magazine

Linux Business and Technology

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. *Linux Business & Technology* is aimed squarely at providing this group with the knowledge and background that will allow them to make decisions to utilize the Linux operating system.

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

Regular features will include:

- Advice on Linux Infrastructure
- Detailed Software Reviews
- Migration Advice
- Hardware Advice
- CEO Guest Editorials
- Recruiting/Certification Advice
- Latest News That Matters
- Case Studies

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE \$71.76

YOU PAY ONLY

\$49.99

12 ISSUES/YR

*OFFER SUBJECT TO CHANGE WITHOUT NOTICE

SUBSCRIBE TODAY!

WWW.SYS-CON.COM

OR CALL

1-888-303-5282

LINUX EDGE
conference & expo

June 3-5 LONDON
June 24-26 BERLIN
September HONG KONG
October CALIFORNIA

FOR ADVERTISING INFORMATION:

CALL 201 802.3020 OR

VISIT WWW.SYS-CON.COM

**SYS-CON
MEDIA**

The World's Leading i-Technology Publisher

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

initely sufficient for mastering the concepts but doesn't facilitate taking it to the next level. I e-mailed Trans-World Resources, LLC, and found out that they can provide more complex code examples at my request.

One cool thing about the provided build scripts is that they're generic. This works great in that the same script can be used to build and deploy the application onto WebLogic 6.x or 7.x, WebSphere 4.x or 5.x, or Oracle9iAS release 2. It would have been nice for the scripts to be Ant based, but because of the amount of material covered in the CBT I was happy to get what was included. For the price of this product and the easy learning that it offered, I was wholly satisfied with what I got.

Support

The only time support is necessary is if the application servers or lab exercises are installed in directories other than those recommended, something I mentioned earlier. In this case, there is clear documentation as to which script files have to be updated with changes. The read-along book comes with a section on troubleshooting, so if you're caught in a bind this might provide enough information to get you through. Otherwise, the book says you can e-mail questions to Trans-World with a 24-hour turnaround time, something I didn't test.

Summary

Overall, Conquer-IT! is a great way to get up to speed on key J2EE skills. The interactive, straightforward nature of the CBT coupled with the practical, real-life exercises makes for a strong alternative to attending a 2-3 day classroom-based training program. On top of it all, the \$349 list price is a bargain. Because the Conquer-IT series is available in Web-based,

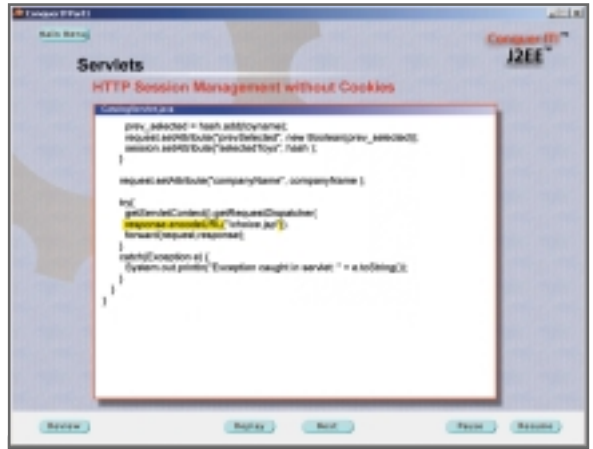


FIGURE 1 Coding analysis section

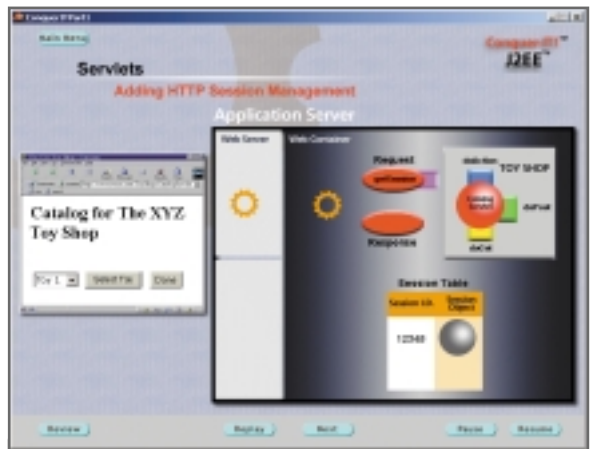


FIGURE 2 Conquer IT! animation

CD-ROM, and instructor-led forms, a blended approach to learning is easily achieved. Conquer-IT! J2EE Part 2: EJB, JNDI, and JDBC is now available. Download a demo from www.twresources.com/coursedemo.asp?leftnav=demo.

About Trans-World Resources

Trans-World Resources offers corporate J2EE training. Their instructors have trained with BEA's corporate clients and at other large corporations including IBM, HP, Cisco, and Oracle. Every attendee of an instructor-led training receives a free full-fledged CBT course as a follow up.

JDJ Snapshot

Target Audience: Good for the Java developer just starting out with J2EE. Not that a more advanced audience would find this too rudimentary to be useful, but it is geared to the J2EE beginner and intermediate developer.

Pros:

- Ease of use
- Takes the complexity out of learning J2EE
- Includes sample build scripts and code examples
- Best practices section

Cons:

- None significant enough to mention



Build Incredible Interactive Graphics
New Release of JGo supports SVG and XML

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

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

**FEATURE
PACKED
NEW
VERSION**

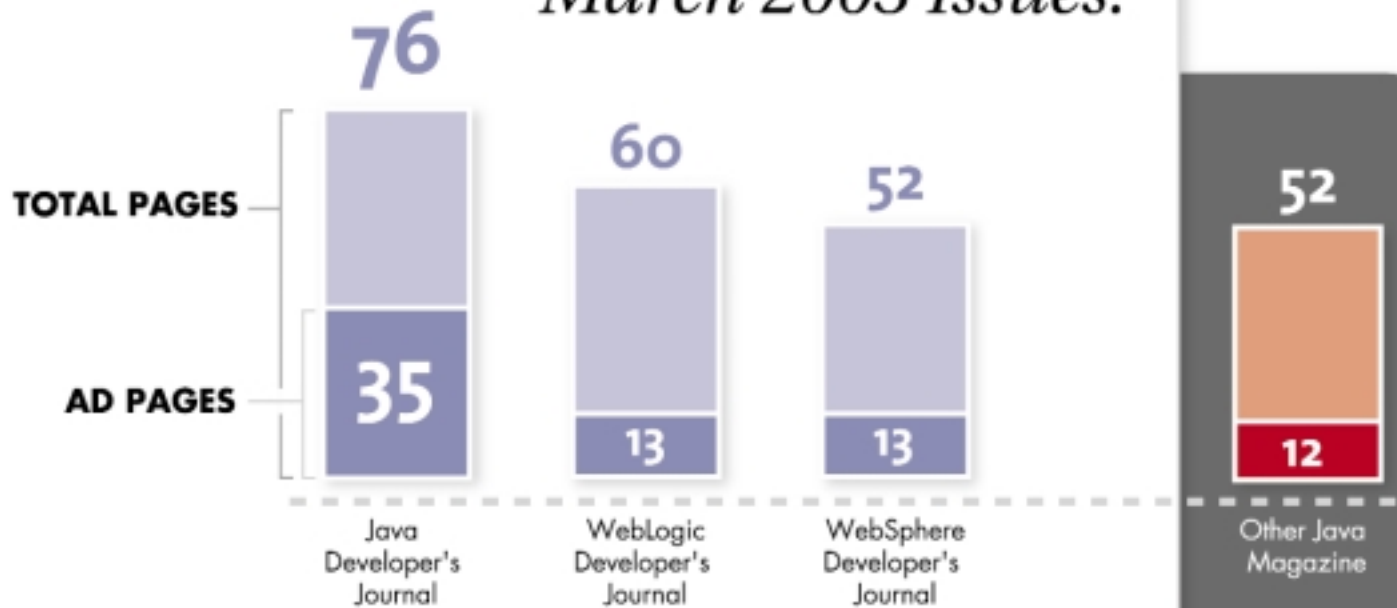
- **FREE Evaluation Kit**
- **No Runtime Fees**
- **Full Source Code**
- **Excellent Support**

For more information, visit
www.nwoods.com/go/
800-434-9820
1-800-886-5780



Java Served Here...

March 2003 Issues:



22%
...of all Java pages in March were published by the other Java magazine publisher

78%
...of all Java pages in March were served by SYS-CON Media



Carmen Gonzalez
Senior VP Marketing



Miles Silverman
VP Marketing



Inside the JCP Process

Eliminating the confusion

WRITTEN BY
ALAN WILLIAMSON

The Java Community Process, or JCP, was formed by Sun in 1998 in response to the community's wish to get involved with the future development of Java. Much has been written regarding JCP, and much confusion exists regarding the whole process and just exactly how much control and influence Sun has over it.

Recently I had the pleasure of meeting with Onno Kluyt, the director of the Java Community Process, who kindly took me through the whole process and answered my questions, which were mostly from some of the misconceptions that run rife in many a newsgroup and mailing list.

Fundamental to the whole program are the members. Members fall into three categories: commercial, government or educational institutes, and individual members. There is a registration fee associated with membership to cover costs, but it's free for individuals and Java licensees. At the time of writing there are approximately 660 members, of which some 220 are individuals. The members hold an election once a year (October/November) to elect two executive committees (EC), one to look after J2EE/J2SE and the other to look after J2ME. Each EC is made up of 16 members, including a permanent seat held by Sun.

This leaves 15 electable seats, of which the five oldest sitting seats are put up for reelection once a year. Once elected the members are "in office" for at least three years. Only JCP members can vote for the EC members and there are no rules as to how many "terms" a member can serve. This is as close to politics as the JCP gets, including the ability for JCP members to put themselves up for election, write a manifesto, and begin lobbying members for their votes. Sun does not manage this voting but hires PricewaterhouseCoopers to manage the whole process, and the five new members are announced during the first week of December.

The EC is responsible for managing the whole JSR process including the approval of new JSRs and ensuring each JSR goes through all the necessary steps,

from initial conception to final approval, and possible absorption into the core Java editions. Another important role of the EC is to ensure that no JSR overlaps with any functionality already provided by existing Java libraries or by JSRs already in existence.

One or more JCP members can propose a Java Specification Request (JSR). This JSR can be a completely new specifi-

The public draft is posted on the Internet for anyone to review and comment on. The spec lead will ensure that the necessary Reference Implementation (RI) and Technology Compatibility Kits (TCK) are available and continually kept up to date. The JSR can stay in the public draft phase for 30-90 days where it's then prepared to be presented to the EC for their final

...the whole process is designed to make sure that everything is kept as fair as possible without the ability to introduce any sort of old-boys network that could ultimately stifle innovation

cation proposal or a significant revision of an existing one. Once submitted, the EC will approve, or deny, the request within 14 days. They want to ensure that the JSR does indeed fit within the bounds of the JCP process and doesn't overlap with any existing JSR or specification. The person submitting the proposal becomes the specification lead who, for all intents and purposes, is the manager of the JSR.

After the JSR has been approved, the spec lead must form an expert group, which is composed of those who are experts in the field proposed by the JSR. The size of this group can be from 6 to 60 members. The expert group's first job is to prepare the community draft of the specification. All members of the JCP can see this draft and channel feedback to the expert group. The community draft goes into review for 30-90 days, and afterward the EC votes on whether or not the JSR is strong enough to go into public draft.

decision, which is achieved through a vote taking no more than 14 days.

If the JSR is successful, it's then taken into the maintenance phase, where the spec lead will keep all the components up to date and continually feed information back to the EC.

There's a lot going on here, and the whole process is designed to make sure that everything is kept as fair as possible without the ability to introduce any sort of old-boys network that could ultimately stifle innovation. The ownership of the final specification can now be licensed under an open-source model should the expert group deem it necessary. However, under the new rules of the JCP, anyone is free to implement the specification as long as they license the TCK for compatibility.

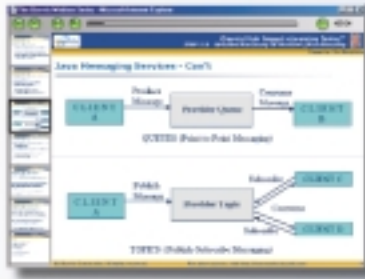
For more information on the whole JCP process, visit www.jcp.org.

alan@sys-con.com

ASK YOUR MANAGER

"Can we purchase these courses for our developers? They are short, high-impact, developed by industry experts, and **just \$29 each!**"

Self-service purchase, instant access ... visit <http://community.isavvix.com> today!



How It Works!

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

Partial List of Courses:

The Struts Framework

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

Getting Started with XSLT

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

J2EE Features and Functionality

This course will provide a summary of the components within the J2EE API. It is designed as an introduction to the extended features of the Enterprise Edition, and perfect for those already familiar with Java who want to understand more about Enterprise Application Development. Concentration will be placed on both client and middle tiers, including Servlets, JSPs, EJBs and JMS, first reviewing the fundamentals then providing code examples for each. An overview of those technologies new to J2EE, including JTA, JCA and Message-Driven Enterprise Java Beans will also be included.

Java Best Practices

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

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

Java Performance Tuning

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

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

For more information, visit <http://community.isavvix.com> or contact us today!

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



My UML Diagrams

Class diagrams on-the-fly from your Java code!



My Dev Space

Free JSP Hosting, XML/XSL, J2SE 1.4, tag lib...

Many more:

Discussion forums, white papers, bookmarks, secure file storage, download links, code sharing, and much more...

About iSavvix

iSavvix is an information-technology consulting and services company that provides custom solutions based on Enterprise Java™, XML and related technologies. Since its inception in 1996, iSavvix has built robust, secure, user-friendly, and cross-platform enterprise applications for many Fortune 1000 and other leading companies.



Delivering Java Technology Solutions for your business

© Savvix Corp. Herndon, VA, +1-888-iSavvix, info@isavvix.com

JDJ Edge 2003 East

INTERNATIONAL JAVA CONFERENCE & EXPO



JDJ EDGE
conference & expo

Boston, MA
March 18-20, 2003

Home



J2EE



J2SE



J2ME



When SYS-CON Media's sister company, SYS-CON Events, began preparing last year for this spring's "Web Services Edge" Conference & Expo, one consideration was paramount: every effort in the nine-month preparation cycle should be geared toward making it indisputably the world's largest independent Java, .NET, XML, and Web services event.

That particular mission was accomplished on March 18-20, 2003, at the centrally located Hynes Convention Center in Boston, Massachusetts, when Web Services Edge 2003 East made its mark right from the get-go, with delegates from a wide variety of companies both technologically and geographically. Not only had they been attracted by the specific session tracks for Java, .NET, XML, and Web services, they had also come to take advantage of the all-day *i*-technology tutorials, whether it was the Sun Microsystems Java University, the IBM XML Certified Developer Fast Path, Russ Fustino's .NET workshop (Russ' Tool Shed), or Derek Ferguson's Mobile .NET tutorial.

The show opened with a very well-attended keynote from Oracle's John Magee, VP of Oracle9i Application Server. Magee stressed that the key to understanding why Web services, unlike its distributed-computing forerunners like COM and CORBA, is prevailing in the enterprise space is that Web services do more than merely enable interoperability between platforms and integration between applications – they also do so simply.

What drives their simplicity, Magee explained to the audience, is standards.

The afternoon keynote offerings on Day One of the conference were equally well received. First came a panel coordinated by the Web Services Interoperability Organization (WS-I). The WS-I is an open

industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages, and the panel discussion took place against the backdrop of the WS-I Basic Profile 1.0, consisting of a set of nonproprietary Web services specifications. The working draft for this, the audience learned, was approved just four weeks before the conference.

But security, the panel agreed, was the primary priority. Now that corporations like Merrill Lynch and DaimlerChrysler have joined the organization, ensuring that everyone adheres to the same specification is more important than ever. Web services is moving beyond mere SOAP, WSDL, and UDDI toward addressing security, messaging, reliability, and transactions. Eric Newcomer, chief technology officer of IONA Technologies, emphasized the importance of the World Wide Web Consortium (W3C) approach to these challenges, an effort that centers on the W3C's Web Services Specification Effort.

The Web services keynote panel was quickly followed by the highlight of Day One for



many of the delegates gathered in the keynote hall: an address by Miguel de Icaza, the impossibly young and extremely gifted founder and leader of the GNOME Foundation, cofounder of Ximian, Inc., and .NET expert extraordinaire – as anyone needs to be who leads a project designed to port .NET to the Linux operating system.

The Mono Project, as de Icaza's project is called, clearly fascinated the broad mix of developers attending the conference.

After explaining that GNOME – a desktop development platform and suite of productivity applications – is his company's key focus and is mostly developed in C, C++, Python, and Perl, he went on to recount how for every new GNOME API (GNOME is component-oriented and supports many programming languages), GNOME developers needed to develop language-specific bindings. Thus .NET, which also addresses the multilanguage problem, was of immediate interest to de Icaza.

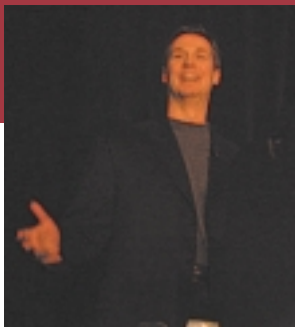
As soon as he learned about the .NET Framework, he told the spellbound audience, he got excited – a single Virtual Execution System for multiple languages, with a large and reusable factored class library, that was, in his view, just what was needed. As well as being a new way to do things, .NET's rich support for interop (COM, P/Invoke) meant you didn't have to rewrite everything all at once.

And so Mono was born: an open-source .NET Framework implementation.

It's based around the CLI ISO standard, de Icaza continued. It has a CLI-compliant execution system and a x86 JIT compiler. It's supported by Windows, BSD, Linux, and Solaris, and there has been lots of progress on the class libraries.

The Windows support, de Icaza said, was merely a function of the fact that 60% or so of Mono developers have a Windows background. Some of the code contributed to Mono was funded by Microsoft grants, he added.

At the end of his keynote address, scores of developers of every stripe got up from their chairs and surrounded de Icaza for further questions. The response to his good humor, rapid delivery, technical savvy, and sheer charm had



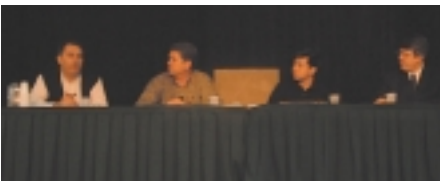
John Magee, Oracle:
"Developing in a Services-Based World"



Mark Herring, Sun Microsystems:
"Bridging the Gap Between WS-Myth and WS-Reality"



Miguel de Icaza, Ximian:
"The Mono Project"



WS-I Panel Discussion:
"A Road Map for Web Services Standards"



Java Panel Discussion:
"The Future of Java"

been overwhelming and with his keynote, Web Services Edge 2003 (East) passed a significant milestone: no previous conference in the series had ever included so wide a range of technical content.

Day Two saw Sun's Mark Herring take the keynote stage and his mastery of the whole Web services paradigm was clearly in evidence. Extended coverage of both his Java keynote and the subsequent keynote address by Jesse Liberty are available on the main conference Web site, www.sys-con.com/WebServicesEdge2003East.

The closing keynote discussion panel, which for many turned out to be the high point of the entire keynote program, was a wide-ranging and a sometimes heated debate about "The Future of Java." The whole intense and highly interactive hour exemplified very well how a SYS-CON *i*-technology conference program differs from that offered by any other conference organizer. This was panel discussion at its best.

True to the enormously close links that *Java Developer's Journal* enjoys with the software development industry, the participants in this final panel at Web Services Edge 2003 (East) had come to Boston from far and wide. Sun's chief technology evangelist Simon

Phipps had flown over from the UK and BEA's director of technology evangelism Tyler Jewell had traveled from Los Angeles. Sonic Software's VP and chief technology evangelist Dave Chappell may have nipped across from Bedford, MA, but Aligo's CTO Jeff Capone had flown in from San Francisco, and JBoss founder Marc Fleury had come up from the JBoss Group's company's HQ in Atlanta, Georgia.

We fully expect the next Conference & Expo, Web Services Edge (West) in October, to be equally chock-full of the movers and shakers of the software development industry as it continues its headlong progress toward distributed computing with full application integration and interoperability.

All in all it was a marvelous conference, and the Expo hall was intensely busy from the moment it opened to the moment it closed two days later.

This is not the end of the Web services "story," nor is it even the beginning of the end; but March 18-20 in Boston's Hynes Convention Center may well have marked the end of the beginning.

Come join us for Phase Two...in October in California. ☘



SUBSCRIBE TODAY TO MULTIPLE

Go To www.sys-con.com/suboffer.cfm



and receive your **FREE CD Gift Package** via Priority Mail



Each CD is an invaluable developer resource packed with important articles and useful source code!

Pick the CDs to go with your Multi-Pack order

▶ Pick one CD with your 3-Pack order

▶ Pick two CDs with your 6-Pack order

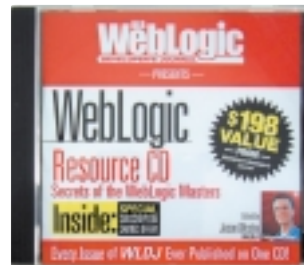
▶ Pick three CDs with your 9-Pack order



More than 1,400 Web services and Java articles on one CD! Edited by well-known editors-in-chief Sean Rhody and Alan Williamson, these articles are organized into more than 50 chapters on UDDI, distributed computing, e-business, applets, SOAP, and many other topics. Plus, editorials, interviews, tips and techniques!
LIST PRICE \$198



The most complete library of exclusive *JDJ* articles compiled on one CD! Assembled by *JDJ* Editor-in-Chief Alan Williamson, more than 1,400 exclusive articles are organized into over 50 chapters, including fundamentals, applets, advanced Java topics, Swing, security, wireless Java,... and much more!
LIST PRICE \$198



The most complete library of exclusive *WLDJ* articles ever assembled! More than 200 articles provide invaluable information on "everything WebLogic", including WebLogic Server, WebLogic Portal, WebLogic Platform, WebLogic Workshop, Web services, security, migration, integration, performance, training...
LIST PRICE \$198



The most complete library of exclusive *CFDJ* articles on one CD! This CD, edited by *CFDJ* Editor-in-Chief Robert Diamond, is organized into more than 30 chapters with more than 400 exclusive articles on CF applications, custom tags, database, e-commerce, Spectra, enterprise CF, error handling, WDDX... and more!
LIST PRICE \$198



The largest and most complete library of exclusive *XML-Journal* articles compiled on one CD! Edited by well-known Editors-in-Chief Ajit Sagar and John Evdemon, these articles are organized into more than 30 chapters containing more than 1,150 articles on Java & XML, XML & XSLT, <e-BizML>, data transition... and more!
LIST PRICE \$198



The most up-to-date collection of exclusive *WSDJ* articles! More than 200 articles offer insights into all areas of WebSphere, including Portal, components, integration, tools, hardware, management, sites, wireless programming, best practices, migration...
LIST PRICE \$198

- Web Services Resource CD
- Java Resource CD
- WebLogic Resource CD
- ColdFusion Resource CD
- XML Resource CD
- WebSphere Resource CD
- Linux Resource CD

Your order will be processed the same day!



An up-to-the-minute collection of exclusive *Linux Business & Technology* articles plus Maureen O'Gara's weekly *LinuxGram*, which keeps a finger on the pulse of the Linux business community. Edited by *LBT's* Editor-in-Chief Alan Williamson, these articles cover migration, hardware, certification, and the latest Linux-related products. Available June 2003!
LIST PRICE \$198

Subscribe Online Today

POPULAR MAGAZINES ONLINE

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

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS



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

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

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

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

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

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

TO ORDER

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

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

<input type="checkbox"/> 3-Pack	<input type="checkbox"/> 1YR	<input type="checkbox"/> 2YR	<input type="checkbox"/> U.S.	<input type="checkbox"/> Can/Mex	<input type="checkbox"/> Intl.
<input type="checkbox"/> 6-Pack	<input type="checkbox"/> 1YR	<input type="checkbox"/> 2YR	<input type="checkbox"/> U.S.	<input type="checkbox"/> Can/Mex	<input type="checkbox"/> Intl.
<input type="checkbox"/> 9-Pack	<input type="checkbox"/> 1YR	<input type="checkbox"/> 2YR	<input type="checkbox"/> U.S.	<input type="checkbox"/> Can/Mex	<input type="checkbox"/> Intl.

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

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

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

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

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

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

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

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

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

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

▶ **Jinfony Releases JReport 5** (Rockville, MD) – Jinfony has announced the availability of JReport 5. JReport now offers the option of running within a clustered environment, providing enterprises with reliability and scalability.

Benchmark results show that JReport 5 has improved performance by as much as 100%. In a clustered environment, multiple servers could be utilized to achieve high throughput. The automatic failover feature ensures that the system will continue to function and deliver reports under any single point of failure. JReport also offers the scalability and the protection against lost data required for enterprise mission-critical applications.

www.jinfony.com

▶ **aJile Brings MIDP 2.0 to Standalone Game Devices** (San Jose, CA) – aJile Systems Inc. has created a game development kit to bring Java MIDP technology to Nintendo's Game Boy Advance players. aJile's JAMiD game development kit uses Sun's latest MIDP 2.0 technology, which offers advanced audio and graphics features to Java mobile game developers.

The key component of the JAMiD kit is the Java game cartridge, which plugs into the external card slot on any Game Boy Advance or Game Boy Advance SP, bringing Java capabilities to the popular mobile game player. The JAMiD cartridge will allow users to run MIDP games while listening to MP3 music.

www.ajile.com

▶ **Quest Software Releases JProbe Suite 5.0** (Irvine, CA) – Quest Software Inc., a provider of application management solutions, has announced the release of JProbe Suite 5.0 with new investigative capabilities and dramatic performance improvements for Java and J2EE developers. JProbe 5.0 combines enhanced memory debugging features with improved speed to help organizations deploy production-ready Java and J2EE applications sooner.

www.quest.com

▶ **Oracle and Sprint Offer Integrated Development Kit** (Overland Park, KS / Redwood Shores, CA) – Oracle Corporation and Sprint have announced integration between

their J2ME environments that will enable Java developers to build mobile applications for the enterprise that can be deployed on PCS Vision devices.

Oracle9i JDeveloper and the Sprint Wireless Toolkit allow Java developers to create, test, and debug J2ME applications using Sprint's J2ME PCS emulators in a unified application development environment. The new extension builds upon the award-winning Oracle9i JDeveloper and its J2EE and Web services capabilities, giving J2ME developers flexibility in constructing mobile enterprise applications.

www.oracle.com

<http://developer.sprintpcs.com>

▶ **Sybase PowerBuilder 9.0 Available** (Orlando, FL) – Sybase, Inc., an enterprise infrastructure and integration company, has released PowerBuilder 9.0. This version of Sybase's rapid application development (RAD) tool enables developers to continue building rich-client applications for mission-critical business requirements, and at the same time propel them further into Web and *n*-tier development with new capabilities. PowerBuilder 9.0 also lays the foundation for "4GLplus", the next RAD environment from Sybase that will bring an even higher level of developer productivity through tight integration of design, modeling, development, deployment, and management.

New capabilities in PowerBuilder 9.0 include RAD JavaServer Page (JSP) Targets, Web services, XML DataWindows, third-party application servers, and the PowerBuilder Native Interface.

www.sybase.com

▶ **CallSource Future-Proofs IT Architecture with BEA WebLogic** (San Jose, CA) – BEA Systems, Inc., an application infrastructure software company, has announced that CallSource, a provider of advertising source tracking, telephone analysis, and training, has selected BEA WebLogic Platform 7.0 as the foundation for its

SUN CREATES JOINT JAVA TECHNOLOGY CENTER

(Hannover, Germany) – Sun Microsystems, Inc., has established a joint Java Technology Center at the SAP AG headquarters in Walldorf, Germany. Building on Sun's leadership and innovation in Java technology, the Center's charter is to assess, promote, and optimize Java technology for adoption in the SAP NetWeaver technology platform, the business-ready, services-oriented technology platform for all SAP solutions.

To support and adopt Java technology in SAP solutions, and to accelerate time-to-market, engineers from both companies will integrate Java technology into SAP solutions, continuously optimize the performance of Java-based applications from SAP, and support SAP customers using Java-based applications, including those running on Windows.

<http://sun.com>

next-generation application infrastructure. CallSource develops customized software and services that enable companies to track and analyze calls, including calls received in response to advertisements in specific media sources.

CallSource plans to leverage BEA WebLogic Workshop to rearchitect its database and develop new applications using XML and Web services for integrated administration, provisioning, online help, and reporting capabilities.

www.callsource.com

www.bea.com

▶ **Nextel and Inciscent Launch IC2** (Falls Church, VA / Reston, VA) – Inciscent, Inc., a wired-to-wireless software and solutions provider, and Nextel Communications, Inc., a provider of integrated digital wireless communications, have announced the commercial availability of Inciscent Command & Control (IC2) network and systems management software for the Blackberry 6510 from Nextel.

IC2, a Java application certified by Nextel for the Blackberry 6510, enables IT professionals to manage, configure, and troubleshoot enterprise information systems and data networks, in real time, from virtually any location on Nextel's nationwide network.

www.inciscent.com

www.nextel.com

ADVERTISER	URL	PHONE	PAGE
Altova	http://jdj.altova.com/authentic5		23
BEA Systems	www.dev2dev.bea.com/useworkshop		7
Canoo Engineering AG	www.canoo.com/ulc/	41 61 228 94 44	19
Computer Associates International, Inc	ca.com/allfusion		9
Crystal Decisions, Inc	www.crystaldecisions.com/cr9/218	800-877-2340	31
Fiorano Software, Inc.	www.fiorano.com	408-354-0846	59
netSoft Technology Corp.	www.inetsoft.com/jdj	888-216-2353	47
Infragistics, Inc.	www.infragistics.com	800-231-8588	14-15
InstallShield Software Corp.	www.installshield.com/puzzlejdi		37
INT, Inc.	www.int.com	713-975-7434	48
Intraware	www.intraware.com/evalapp/	800-469-4313	35
iSavvix	http://community.isavvix.com	703-689-3190	67
Interactive TKO, Inc	www.itko.com/lisa		11
JDJ Workshop	www.sys-con.com/education	201-802-3058	61
Jinfonet Software	www.jinfonet.com/j4.htm	301-838-5560	33
Linux Business & Technology	www.sys-con.com	888-303-5282	65
Macromedia	www.macromedia.com/go/jrun4jdi	415-252-2000	55
Northwoods Software Corp.	www.nwoods.com/go	800-434-9820	64
ObjectFocus, Inc.	www.objectfocus.com	408-973-8200	63
Oracle	oracle.com/experts	800-633-1072	17
Parasoft Corporation	www.parasoft.com/jdj2	888-305-0041	21
Precise Software	www.precise.com/jdj	800-310-4777	25
QUALCOMM Incorporated	www.brew2003.com		51
Quest Software, Inc.	http://java.quest.com/jclass/jdj		41
Quest Software, Inc.	http://java.quest.com/performance/jdj		57
Quest Software, Inc.	http://java.quest.com/jprobe/jdj		Cover IV
Rational Software	www.ibm.com/rational		27
ReportingEngines	www.reportingengines.com/download/jdj3/jsp	888-884-8665	28-29
ReportingEngines	www.reportingengines.com/download/jdj2/jsp	888-884-8665	53
ReportMill Software	www.reportmill.com/webstart	214-513-1636	45
Salmon LLC	www.salmonllc.com/sofia-jdj		43
Sonic Software	www.sonicsoftware.com/jdj		Cover II
SpiritSoft	www.spiritsoft.com/climber		Cover III
Swingtide, Inc.	www.swingtide.com/testdrive		13
SYS-CON Subscription Offer	www.sys-con.com/suboffer.cfm		70-71
WebAppCabaret	www.webappcabaret.com/jdj.jsp		49
Zero G	www.zerog.com	415-512-7771	3

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

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

What's in the next issue of *JDJ*?

JAVA SERVER FACES

Developing interesting and effective Java Web applications requires simple, robust, and manageable frameworks and tools that complement them. If you design and develop Java applications for a living, it could be quite a challenge to stay abreast of all the software developments and frameworks both from commercial software vendors and the vast open-source community. This article surveys the various Java Web development frameworks that are popular today and then takes a deeper look into the Java Server Faces (JSF) specification.



ARE YOU USING ABSTRACT CLASSES, POLYMORPHISM, AND INTERFACES?

If the answer is no, your project at a minimum needs a code review. Using abstract classes, interfaces, and polymorphism will improve the design of your project by making it more readable and easily extensible. In addition, it will make your code more compact and elegant.



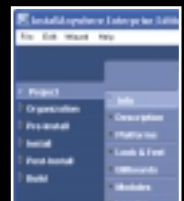
TRIMMING THE FAT FROM SWING

I'm sure we have all heard it before: Java on the client is slow; Swing is slow. The reality is that Sun has made great progress in increasing the speed of Swing and Java on the client. However, we developers must do our part to show that Java has indeed improved to the point of usability and viability on the client.



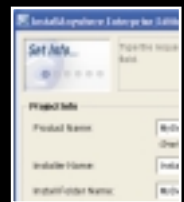
FIRSTSQL/J PROFESSIONAL V2.0 BY FIRSTSQL

A well-designed database offers flexibility, speed, and easy data manipulation through JDBC support. This database must be 100% Java to maintain cross-platform compatibility as well as easy to use so the average Java programmer can integrate it with little difficulty. FirstSQL provides such a database in FirstSQL/J Professional.



INSTALLANYWHERE 5 ENTERPRISE EDITION BY ZEROG

Creating cross-platform applications has always been a challenge. Java can insulate us from much of the hassle during application development, but often the particulars of each platform become painfully acute when distributing the application to a host of varied platforms. InstallAnywhere from ZeroG is a Java-based install product that allows you to create and customize cross-platform installers for any type of application.



The McNealy Windscale of Performance Benchmarks



One of the recent stirrings to occur inside the Java industry is what has become known as “memogate.” A Sun engineer wrote an internal memo to his colleagues that listed a number of problems with Java on Solaris, ranging from large VM footprints and issues with serialization to the usual salvo of complaints about Swing.

While the memo was intended to be read internally, the inevitable occurred and it ended up on the Internet (www.internalmemos.com/memos/memodetails.php?memo_id=1321) and generated some pretty big shock waves in the computing press. The fact that it was an internal memo criticizing the implementation of Java on Solaris meant it was a double-edged whammy that attacked Sun from behind their defenses. The legal boys at Sun had nothing in their training to prepare them for this kind of situation. What were they to do?

Two days later Scott McNealy spoke at a product launch. The inevitable question about memogate was asked, and the gathered press held their breath to await the response. What pearls of wisdom, subtle diplomacy, or words of comfort could save Solaris and Java, both sunk from the same rogue imploding torpedo. “It runs Java like the wind” came the simple reply (http://news.com.com/2100-1001-984529.html?tag=fd_top). The following day the issue had literally blown away. A new manifesto had been written, a new battle cry given, a new dawn for the Californian Sun.


Exactly how fast is the wind? Is running like the wind better than running like a dog?

Back in 1806 a man called Francis Beaufort categorized wind speeds in 12 forces, each one representing a sea condition; then updated to represent a land condition as well. 0 is calm where the “smoke rises vertically.” 6 is a strong breeze characterized by “large branches in motion,” and 12 is hurricane force with “widespread damage.” Perhaps Java needs its own scale of windy euphemisms, named the McNealy Scale in honor of the great orator himself. ☼

FORCE	DESCRIPTION	CONDITION
0	Java 2 bundled with Windows OS	Outlook poor to medium
1	Only one company can be the market leader in the J2EE app server space	Difficult to see which way the wind is blowing. Lots of hot air from both sides as analysts sway unpredictably from side to side.
2	The years behind WebLogic that BEA claims WebSphere is	Following the felling of huge forests to print new specifications, the response time of the boys in blue to arrive on the scene has declined from previous years.
3	Sun stock price	Current long-term low pressure over Santa Clara. High pressure not imminent.
4	Trying to use container-managed EJBs to build J2EE apps	Regrouping of the component broker thunderstorms of the mid '90s that flooded numerous project passages, making them impassable to teams that became bogged down in architectural mud and failed to perform.
5	What Scott Violet at Sun thinks of SWT	Trees and tables start falling over, and moving forward through large lists is difficult.
6	What Bernie Spang at IBM thinks of Swing	Visibility poor. Unable to make out basic shapes as edges and objects become blurred and difficult to identify.
7	...and a half years since Java was first launched	Wispy clouds, choirs of angels, and shafts of heavenly light radiating toward James Gosling's general direction. We are not worthy.
8	Seconds it takes the light from the Sun to reach the Earth	Something Sun's marketing department needs to try and emulate when they have to respond to Microsoft FUD about “impartial” .NET versus J2EE comparisons.
9	Original Wali Songo missionaries who were canonized by the Indonesians on the island of Java	Difficult to keep a grasp on relevant ideas as the deadline approaches; begins to blow magazine from editor's hands.
10	Commandments of Java that were broken by Bill Gates	Friendly looking clouds can quickly develop into tornadoes trying to suck up everything in their path. “We believe in Java the language, but not Java the religion” – Bill Gates, June 17, 1997.
11	Number of Spanish mobile phone carriers who deploy J2ME application servers	Cool fronts arriving from the east and west will merge over the central U.S., resulting in some very exciting phenomena to watch. Bring protective eyewear.
12	Runs like the wind	Java on Solaris



Credit: Jason Chambers



No Room for Error

Application integration is an intimidating challenge for any enterprise; the downtime costs of unreliable integration are unthinkable 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

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



For more info and a free eval, visit:

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