

JAVA™ DEVELOPER'S JOURNAL

JavaDevelopersJournal.com

Volume: 2 Issue: 7

The Risks from Java

by Linda Garceau & Victor Matos pg. 5

Without Policy, "Security" Isn't Really Secure

Gary E. Brooks pg. 7

Ask Dr. Java

A Little of Everything

by Jerry Ablan pg.50

Tips & Techniques

PlumbingJavaClasses

by Brian Maso pg.40

Under the Sun

Data Visualization

pg.44

The Softer Side

The Grind

by Joe S. Valley pg.66

Product Reviews

Net Dynamics

by Khanderao Kand pg.60

Roaster Release 3

by Jim Mathis pg.32

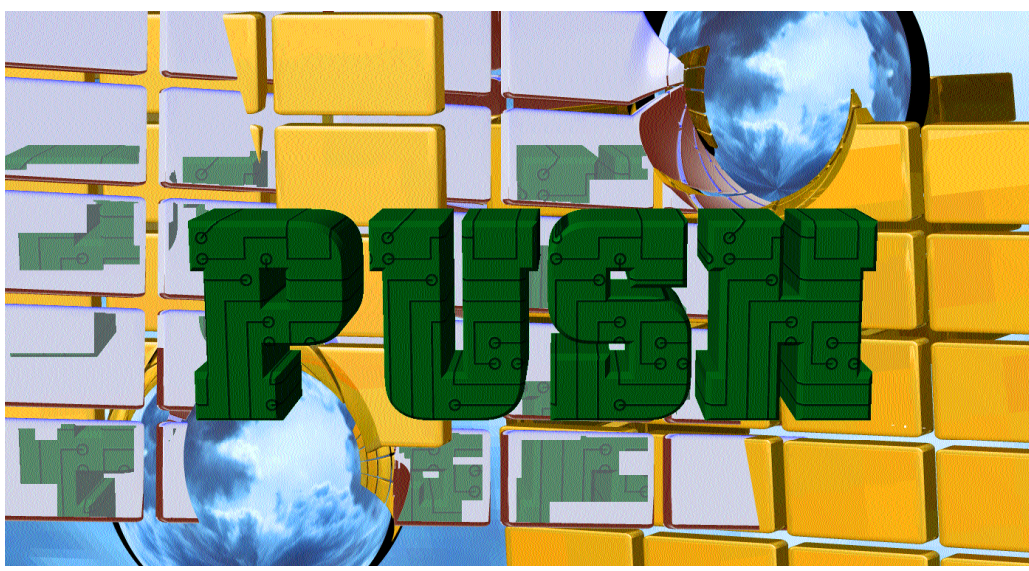
PowerJ

by Jeff Winchell pg.56

CORBACorner

RMI on CORBA

by Jeff Nelson pg.46



JDJ Feature: Invasion of the Intranet

John Fronckowiak

Using Internet technology to share corporate information

8

JDJ Interview: Turning Success at Home Into Revenue Abroad

Andrew Raskin

Applix, Inc. joins forces with K.K. Ashisuto in the Japanese Market

28

JDJ Security: Making Java More Secure

Jay Heiser

Upgrading the runtime environment to reduce the risks

36



JDJ Feature: Web Database Publishing

Siet-Leng Lai & Joo Hwee Lim

Injecting liveliness and information into the Web

20

JDJ Feature: Extending the AWT

Elie Levy

Creating your own components—an Image Button

52



Visual J++ Corner: Application Foundation Classes

John Fronckowiak

More user friendly controls for your applications

16

NEW COLUMN Visual Café Pro

Alan Williamson

Looking at one of the most popular IDEs in use today

42



Web Logic Full Page Ad

Schlumgerger Full Page Ad

Bristol Full Page Ad

EDITORIAL ADVISORY BOARD

Randy Cook, Ted Coombs, Allan Hess,
Jack Hidary, Arthur van Hoff, Kim Polese,
Brian Maso, Richard Soley, David Spenhoff

Art Director: Jim Morgan
Executive Editor: Scott Davison
Managing Editor: Gail S. Schultz
Copy Editor: Alix Lowenthal
Technical Editor: Randy Cook
Java for the Mac Editor: Luke Cassady-Dorion
Class Libraries Editor: Claudia Piemont
Product Review Editor: Jim Mathis
Games & Graphics Editor: Eric Ries
Tips & Techniques Editor: Brian Maso
Security Editor: Jay Heiser

WRITERS IN THIS ISSUE

Tony Davis, John Fronckowiak, Jay Heiser, Qusay
Mahmoud, Brian Maso, J.P. Morgenthal, Jeff Nelson,
Judy Rawls, Keith Rowe, Jon Steiner,
Joe S. Valley, Alan Williamson, Ed Zebrowski

SUBSCRIPTIONS

For subscriptions and requests for bulk orders,
please send your letters to Subscription Department

Subscription Hotline: 800 513-7111

Cover Price: \$4.95/issue.

Domestic: \$49/yr. (12 issues)

Canada/Mexico: \$69/yr.

Overseas: Basic subscription price plus air-mail postage
(U.S. Banks or Money Orders).

Back Issues: \$12 each

Publisher, President and CEO: Fuat A. Kircaali
Vice President, Production: Jim Morgan
Vice President, Marketing: Carmen Gonzalez
Marketing Director: Larry Hoffer
Accounting: Fran Giannattasio
Senior Designer: Robin Groves
Web Designers: Robert Diamond
Customer Service: Patricia Mandaro
Rae Miranda
Erin O'Gorman
Editorial Intern: Anita Dawn Fee
Advertising Intern: Shannon Dowd

EDITORIAL OFFICES

SYS-CON Publications, Inc.
39 E. Central Ave., Pearl River, NY 10965
Telephone: 914 735-1900 Fax: 914 735-3922
CompuServe: 73611,756

Postmaster: Second class postage is paid in
Jersey City, NJ and other USPS locations.
Send address changes to Editorial Offices.

© COPYRIGHT

Copyright © 1997 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, SYS-CON Publications, Inc. reserves the right to revise, republish and authorize its readers to use the articles submitted for publication.

ISSN # 1087-6944

DISTRIBUTED in USA by

International Periodical Distributors

674 Via De La Valle, Suite 204, Solana Beach, CA 92075 619 481-5928

BPA Membership Applied For August, 1996

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.



J. P. Morgenthal



Money from Java

Ka-Ching! That's the sound of Sun's cash registers ringing every time a cellular phone, pager, set-top cable box or host of other electronics equipment is sold. Java is an excellent enterprise software platform. It is the first real competition to the Wintel dynasty in over ten years, but Java is about much more than enterprise software development. In fact, Java is about only enterprise software development as much as a Corvette is about a Sunday afternoon drive.

Embedded Java is a term used to represent a Java Virtual Machine that is stored permanently in silicon. The Virtual Machine is a machine instruction processing unit not so different from the Intel Pentiums that run many of today's PCs. Yet, this Virtual Machine was developed for embedding in microprocessors that would go into consumer electronics equipment. These embedded Java VMs come as chips that will be used in everything from smart cards to your automobile. Don't get the wrong impression; embedded microprocessors have been used for years in electronics equipment, but this is the first standardized machine instruction set that will be used consistently. The reason for this popularity is two-fold:

1. Due to its architecture, Java Virtual Machines can produce extremely high performance.
2. There is a fast growing base of development resources that will be able to develop software for it.

Today, extremely technical C and Assembler programming language gurus develop embedded applications with the knowledge that they will be stored in chips. This requires the ability to optimize the code through both programming and compilers. However, with a high-performance Java chip installed in the device, any Java programmer can develop the same application at a lower cost.

The number of specialized chips developed every year for embedding into electronic devices far exceeds the number of desktops sold over a ten year period. Now, imagine that JavaSoft gets just \$.20 for each device that has a Java Virtual Machine embedded inside of it. Add to that price the cost of the electronics equipment that uses Sun's own Java chips. While this price is only speculative for purposes of visualiza-

tion (Sun has released no pricing information for this type of license yet), the income from this venture alone is staggering.

We've already mentioned consumer electronics as one outlet that will generate tremendous income from per unit sales; another is manufacturing. Manufacturing is an interesting opportunity for income from Java licenses. Nothing would be more beneficial to this industry than being able to have a single development environment for modeling, simulation and assembly. Indeed, no group has struggled more to have a common representation that could be used for all aspects of manufacturing. Java is the first opportunity to possibly achieve this goal since the adoption of STEP, a standard representation of manufacturing-based data. If Java were adopted for this goal, it would require Virtual Machines inside of every single Computer-Aided Design (CAD) application down to inside of every piece of specialized shop floor equipment. Sun would receive income from every license for every Java Virtual Machine inside of this environment.

If, after reading this editorial, you're still unsure why Sun has spent millions developing and pushing an "open" Virtual Machine platform, consider this: Andrew Carnegie made his fortune by owning railroads and the steel to build the trains and the tracks, along with the early telecommunications system to send messages between stations. Sun's product offerings are as cohesive as Andrew Carnegie's. They have an excellent hardware and software offering for enterprise computing. And they own the chips that go in the machines. And they own the patent on the Virtual Machine running inside each of those chips. And it all comes back to Java! ●

Note: A copy of this editorial was sent to Sun Microsystems for their comments. At press time, none had been received

About the Author

J. P. Morgenthal is an industry analyst and a truly independent voice in this industry. His firm, nc.focus, formerly New Horizon Computing Corp., focuses mainly on Java and network computing, delivering industry trends and practical business solutions to complex IT problems. More information regarding nc.focus can be found at www.iffa.com.



jeffreym@iffa.com

Suntest Full Page Ad

**PHONE & ADDRESS
DIRECTORY****CALL FOR SUBSCRIPTIONS
1 800 513-7111**International Subscriptions
& Customer Service Inquiries
914 735-1900
or by fax: 914 735-3922E-Mail: Subscrib@SYS-CON.com
<http://www.SYS-CON.com>MAIL All Subscription Orders or
Customer Service Inquiries to:**JAVA JOURNAL**
DEVELOPER'SJava Developer's Journal
<http://www.JavaDevelopersJournal.com>**Delphi Journal**
PROGRAMMER'SDelphi Programmer's Journal
<http://www.DelphiJournal.com>**PowerBuilder Journal**
DEVELOPER'SPowerBuilder Developer's Journal
<http://www.PowerBuilderJournal.com>**Sybase Journal**
DEVELOPER'SSybase Developer's Journal
<http://www.SybaseJournal.com>Secrets Of The PowerBuilder Masters
<http://www.SYS-CON.com>SYS-CON Publications, Inc.
39 E. Central Ave.
Pearl River, NY 10965 - USA**EDITORIAL OFFICES**
Phone: 914 735-1900
Fax: 914 735-3922**ADVERTISING & SALES OFFICE**
Phone: 914 735-0300
Fax: 914 735-7302**CUSTOMER SERVICE**
Phone: 914 735-1900
Fax: 914 735-3922**DESIGN & PRODUCTION**
Phone: 914 735-7300
Fax: 914 735-6547**THE WORLD TOUR WITH PBDJ
800 513-7111**DISTRIBUTED in the USA by
International Periodical Distributors
674 Via De La Valle, Suite:204
Solana Beach, CA 92075
Phone: 619 481-5928

J. P. Morgenthal



Money from Java

Ka-Ching! That's the sound of Sun's cash registers ringing in the money every time a cellular phone, pager, set-top cable box or host of other electronics equipment is sold. Java is an excellent enterprise software platform. It is the first real competition to the Wintel dynasty in over ten years, but Java is about much more than enterprise software development. In fact, Java is about only enterprise software development as much as a Corvette is about a Sunday afternoon drive.

Embedded Java is a term used to represent a Java Virtual Machine that is stored permanently in silicon. The Virtual Machine is a machine instruction processing unit not so different from the Intel Pentiums that run many of today's PCs. Yet, this Virtual Machine was developed for embedding in microprocessors that would go into consumer electronics equipment. These embedded Java VMs come in the form of chips that will be used in everything from smart cards to your automobile. Don't get the wrong impression; embedded microprocessors have been used for years in electronics equipment, but this is the first standardized machine instruction set that will be used consistently. The reason for this popularity is two-fold:

1. Due to its architecture, Java Virtual Machines can produce extremely high performance.
2. There is a fast growing base of development resources that will be able to develop software for it.

Today, extremely technical C and Assembler programming language gurus develop embedded applications with the knowledge that they will be stored in chips. This requires the ability to optimize the code through both programming and compilers. However, with a high-performance Java chip installed in the device, any Java programmer can develop the same application at a lower cost.

The number of specialized chips developed every year for embedding into electronic devices far exceeds the number of desktops sold over a ten year period. Now, imagine that JavaSoft gets just \$.20 for each device that has a Java Virtual Machine embedded inside of it. Add to that price the cost of the electronics equipment that uses

Sun's own Java chips. While this price is only speculative for purposes of visualization (Sun has released no pricing information for this type of license yet), the income from this venture alone is staggering.

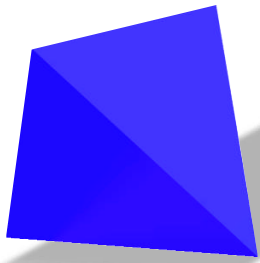
We've already mentioned consumer electronics as one outlet that will generate tremendous income from per unit sales; another is manufacturing. Manufacturing is an interesting opportunity for income from Java licenses. Nothing would be more beneficial to this industry than being able to have a single development environment for modeling, simulation and assembly. Indeed, no group has struggled more to have a common representation that could be used for all aspects of manufacturing. Java is the first opportunity to possibly achieve this goal since the adoption of STEP, a standard representation of manufacturing-based data. If Java were adopted for this goal, it would require Virtual Machines inside of every single Computer-Aided Design (CAD) application down to inside of every piece of specialized shop floor equipment. Sun would receive income from every license for every Java Virtual Machine inside of this environment.

If, after reading this editorial, you're still unsure why Sun has spent millions developing and pushing an "open" Virtual Machine platform, consider this: Andrew Carnegie made his fortune by owning railroads and the steel to build the trains and the tracks, along with the early telecommunications system to send messages between stations. Sun's product offerings are as cohesive as Andrew Carnegie's. They have an excellent hardware and software offering for enterprise computing. And they own the chips that go in the machines. And they own the patent on the Virtual Machine running inside each of those chips. And it all comes back to Java! ●

About the Author

J. P. Morgenthal is known worldwide as an industry analyst and is one of the few truly independent voices in this industry. His firm, nc.focus, formerly New Horizon Computing Corp., focuses mainly on Java and network computing, delivering industry trends and practical business solutions to complex IT problems in this area. More information regarding nc.focus can be found at www.iffa.com.

jeffreym@iffa.com



Bringing your Web site to your
customer – giving them only
what they want to see

Creating a Push Strategy for your Web site

by Keith Rowe

Introduction

What if you built a Web site and nobody came? Or they came once and forgot to check in again to see all the latest stuff you built? One of the problems with the Web is that it has been hard to get people to become regular users of your site. You're either on the Favorites List or you're not. And even if you're there, there's no promise that people will think to call up your site each day.

Push technology allows you to reach back out to individual users and give them content without them remembering to come and look for it. If you do it right, you'll also remind them to come back to see the rest of the great stuff you've been working on.

Push technology has been around for a few years on the Internet. PointCast delivered the first broadly successful solution in 1995. They created a custom Internet application that ran on the user's machine and pulled material off the server. But the system was proprietary and there was no way for you to participate without working directly with PointCast.

Now, the leading browser vendors are bringing the power of Push to site producers everywhere. Both Internet Explorer 4.0 and Netscape Communicator have great solutions that allow you to deliver your content directly to those who want it.

Push technology comes in a number of different forms. At MSNBC, we've been experimenting with a number of these technologies for the last year. In this article, I'll share some of what

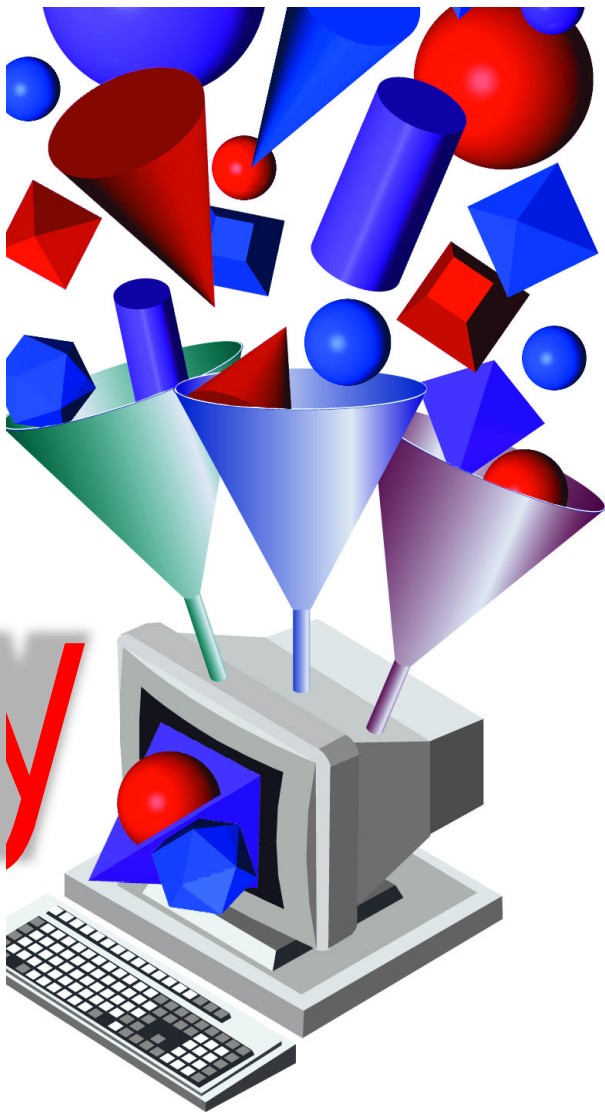
we've learned and provide some tips and tricks for making Push technologies work for you on your Web site.

Personalized Content

Push technologies are very tempting for site producers. You have so much to tell customers that you just know they are dying to hear about. The idea of a system that automatically forces your message down your customer's throat – willing or not – seems like a great idea. That's why telemarketers have so many companies paying them to call you when you're at dinner.

But before you interrupt the user's day with your message, you'd better have something interesting to say. Users quickly tire of interruptions that are irrelevant to their needs. That's why we've focused so much attention on building elegant personalization into our site. Users can easily tell us what their preferences are and we can use that information to deliver only the news they want to see.

MSNBC centers its personalization on the Personal Front Page. This is a server-side script page that assembles the user's choice of leading news stories from around the world, sports scores, stock tables, weather and local news. Users can customize this page to their preferences. We use the Active Server Page (ASP) system in Microsoft's Internet Information Server to build these pages for each user. The ASP script checks a cookie on the user's machine to find the correct customization settings. These set-



things are kept on the server in a SQL Server database. The ASP script then assembles a page of HTML from a series of "mini-pages" and text files to produce the final Personal Front Page.

A Range of Delivery Options

Push technologies are intrusive. To a greater or lesser extent, they interrupt the user's daily work flow and try to grab attention. We manage these interruptions by creating a range of tools with different levels of intrusiveness.

One way to characterize these tools is by how much they interrupt the user:

- Low – The site can construct pages on the fly that appear when the user chooses to come to the site.

once they've decided to come to your site.

- Medium – Sites can send e-mail. This will wait in the user's Inbox until they choose to look at it. This level also includes some forms of cached content where the material is read from the site and kept on the users' local hard drive until they are ready to read it.
- High – There are separate applications that constantly scan for information and interrupt the user immediately when something interesting happens "out there".

We give the customer the ability to select the tools they want to use and the opportunity to personalize the news that is delivered through each of those tools. Delivery options and Personalization work together to provide the customer with the highest value for the interruptions that we create.

Let's look at these delivery options in more detail.

Cached Content

Push technology is driven by cached content. Content is read from a server and stored on the user's local machine. Then the viewing tools can spend more time on formatting and presentation. PointCast is the most common early example of this.

Last year, MSNBC built a similar system

– MSNBC News Off-line – that relies on HTML for the presentation. The user loads a custom client application and then can select what types of stories to keep current. These choices are managed down to the sub-category level (e.g., golf and baseball within the broader category of Sports). The client runs in the background and loads new HTML pages from the site on a schedule selected by the user. Instead of getting the standard front pages authored by the editorial staff, MSNBC News Off-line presents a view by category, sub-category and story.

This system has been in place for seven months. It is about to be supplanted by a more general implementation provided by the upcoming Internet Explorer 4.0 release. IE 4.0 manages an off-line cache based on the notion of channels. Sites can easily author a channel by maintaining a single index file. This index file uses the new Channel Definition Format to describe the parts of the site that should be downloaded to the user's local hard drive. The Channel Guide then allows the user to see when new material is loaded and easily switch among the channels to which they have subscribed.

The IE 4.0 Channel Guide model has advantages for both the site producer and the end user. The producer doesn't need to maintain a separate tool or do much unique authoring for the system. Users have a single model that organizes all the content they care about the most and allows them to browse it quickly.

Personal Delivery

Many sites already use mailing lists to "get the word out" about their site. Pathfinder, the LA Times, Berkley Systems and Microsoft's Slate magazine are all examples of sites that use standard e-mail list servers to tell users when material changes on the site. The advantage of this approach is that it's easy to set up. It's also easy for the user to ignore. Text-based e-mail doesn't provide much opportunity to grab the user's attention. Also, since the material is usually simple promotion, users quickly learn to ignore them unless they are deeply interested in the material.

MSNBC has a different approach to using mail, called Personal Delivery. We built a



Figure 1: IE 4.0 users have one button access to MSNBC.

series of proprietary server-based solutions to deliver individual HTML pages to each user at the time they want it. A scheduling application keeps track of users who have signed up for the service and when they want to receive their pages. At the appointed time, a second server application builds the HTML page, encodes it as a standard MIME e-mail attachment and mails it to the user. The user is given a custom decoder application that reads the attachment, breaks it into a series of HTML pages and opens the browser of choice on the page.

The advantage of this solution is that it brings real HTML-based content to the user directly through the tool many people use the most on the Internet – e-mail. The material is pulled directly off the real Web site so it is current and doesn't require a separate editorial process to maintain. Also, since the entire page, including graphics, is encoded, users can read their news even when they are disconnected from the network. Since it uses regular MIME attachments, it works with any modern e-mail reader. In the next year, many e-mail readers will handle HTML directly so the extra encode/decode step will not be necessary.

One disadvantage of this solution is that it requires extensive server code to implement. Internet Explorer 4.0 provides an easy way to replicate almost the same behavior as part of a Channel implementation. The site producer can set which pages from the site should be e-mailed directly to the user. This is handled on the client side so that no additional server software is needed.

We give the customer the ability to select the tools they want to use and the opportunity to personalize the news that is delivered through each of those tools.

News Alert

MSNBC News Alert is a small Win32 application. It runs all the time on the user desktop and every ten minutes (or at an interval selected by the user) goes out to the Web. It checks the timestamp on a text file that we maintain that is an index of all the stories currently on the site. If the file

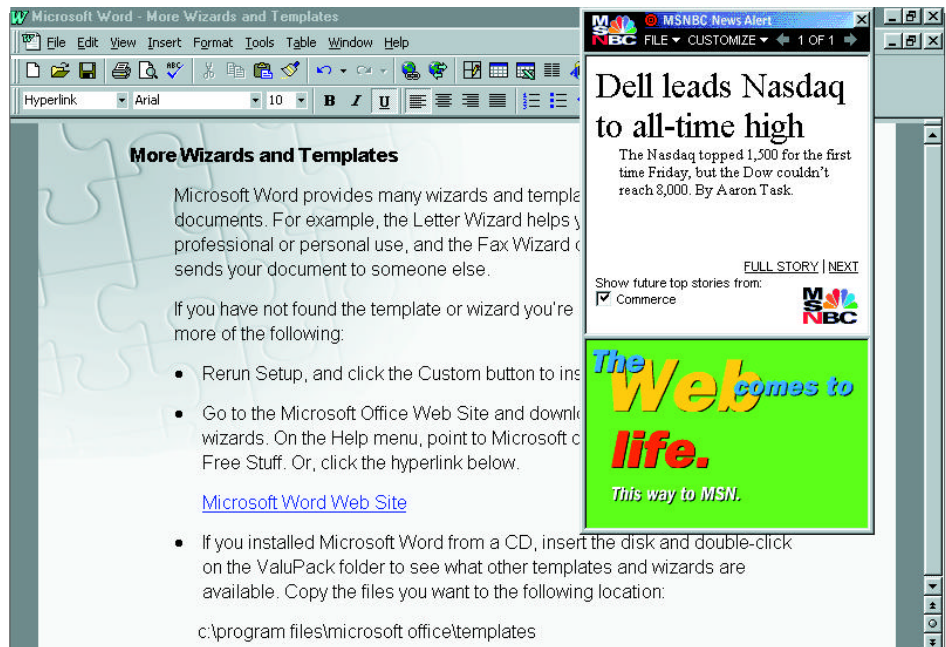


Figure 2: MSNBC News Alert instantly notifies users of breaking news.

has changed, it copies it to the user's machine and scans it for stories users have said they care about.

When a story is found, News Alert triggers an animated icon in the Win95 Task Bar. If the user clicks on the animated icon, a small window appears with the headline. The user can dismiss the box, cycle through any other headlines that News Alert downloaded or just click on the headline itself to go on to the Web to see the full story.

The stories that appear in News Alert are selected in three different ways. MSNBC editors select the top stories in each section of the site all day long. By default, News Alert brings these headlines to the user. Even more interesting, editors have a tool that allows them to immediately place a breaking news headline into the central index file. Users who have News Alert running on their machines will see that headline at their desks the next time News Alert checks the file. This allows us to deliver breaking headlines as soon as they happen, instead of just when the user remembers to check the site.

The Microsoft Investor Web site uses News Alert to announce major moves in stock prices to their users. The user selects the stocks they want to track and the percentage movement that is significant to them.

News Alert also delivers an ad with each alert delivered. News Alert loads the ad in the background and reports back how many times the alert ad was presented. MSNBC sells the spot on a daily or weekly basis to premium advertisers who are looking for new ways to capture attention. In this way, News Alert draws users to the site and creates its own revenue stream.

Tickers

MSNBC maintains a ticker of the latest sports scores and sports headlines. This ticker appears in the MSNBC Sports section and, if the user has Internet Explorer 4.0, it can be pulled off the Web page and placed directly on the user's desktop. This same engine will be used for news and business headlines in the future.

For portability, this ticker is implemented in Java. The applet makes a straightforward get call to a fixed URL. The URL is a server-side script that accesses a SQL database which receives a constant stream of updated scores. The server-side script builds an HTML page with all the latest scores. The Ticker applet reads and formats these scores.

Think Like the User

Now let's talk about how to put these all together.

One of the first rules in any site design is to think like the user. What does your customer want and need? Because you are so close to your site every day, it's easy to imagine that your users want everything dropped right onto their desktop. After all, you went to a lot of trouble to produce that material, shouldn't they do you the courtesy of studying it all? You have to take a more detached view of your product and understand it from your user's point of view. Build a model of your users' behavior. How often do they come to the site? Once a day? Once a week? Once a month? Do they return regularly? How many pages do they look at? What do they look at? There are a number of tools on the market that can interpret your site logs and give you detailed information on what your users

RogueWave Full Page Ad



Figure 3: MSNBC Personal Desktop Tickers sit on users' desktops tracking news and information.

are doing. We've used these tools along with regular surveys on the site to gather an understanding of how our users behave.

Now, imagine what information you have that the user most wants. How often does that information change and update? How urgent is that change to the user? Stock prices are an obvious example of changes that are urgent to some users, but even the most dedicated stock follower can't be distracted by every hiccup in hundreds of stocks. Determine what level of information and what degree of significance it will have for your user.

We've designed our most intrusive applications, like News Alert, to deliver quick snips of information that are truly important. This is both a technical and editorial issue. The editors must have the discipline not to mark every new story as a breaking headline. We've found that when we interrupt with anything less than truly breaking headlines our users complain. We've also found that customers love to take control of the News Alert application and set their own keywords to tune the headlines to their own tastes.

Think Like the Producer

Successful product design starts with the user in mind. But still, you have to remember what the goals of your organization are as well. As a news site, MSNBC feels strongly that there are some stories that people just need to hear about. We reserve the top part of our Personal Front Page for our editors to lead with the top five stories of the day. Sometimes these are news-driven headlines and sometimes they are the most interesting and elaborate features we've built that day. By combining this editorial sensibility with the user-driven personalization, we build a product that is tuned to the user and still has the unique voice of our editorial staff.

At MSNBC, we're trying to run a business. We get most of our revenue from advertising. All this Push technology is wasted if it doesn't somehow drive up these advertising revenues. In some cases, the relationship is very obvious: We sell a special advertising slot with News Alert that runs at a premium because it comes up when headlines are breaking. Most of the increase in revenue comes from

increased customer loyalty, increased return visits and our ability to drive people from the push-delivered news back to the full site.

Summary

Until now, it took a strong development team to design, implement and maintain a push technology solution for your Web site. With the latest round of browsers, these tools are now available to sites of all sizes. Therefore, it is now more important than ever that a solid strategy accompany the use of this technology. ●

About the Author

Keith Rowe is the Director of Technology at MSNBC on the Internet. He has worked at Microsoft Corp. for seven years, starting in Language Design. In 1994, he transferred to lead the prototyping and development efforts for Microsoft's interactive television applications. As Director of Technology, Keith is in charge of the creation of tools and platforms for creating and distributing MSNBC's new products on the Internet. Keith can be reached at keith.rowe@msnbc.com



keith.rowe@msnbc.com

Assymetrics Full Page Ad



AFC – User Interface Controls

Beautify your Java application interfaces with the Application Foundation Classes.

by John Fronckowiak

Foundations

This month I wrap up my two part review of the Application Foundation Classes with a closer look at the user interface controls it provides. In my past two columns, I've showed you how to install and use the Java SDK 2.0, which includes the Application Foundation Classes, and how to begin using the Application Foundation Classes in your own applications. Now it's time to take a closer look at what the Application Foundation Classes can do for the user interfaces of your Java applications.

The Controls

The Application Foundation Classes provide a number of enhanced user interface controls which can be used in your applications. These controls include:

- Static controls – Used to display static text, graphic or both
- Buttons – Used to create push buttons, radio buttons and check boxes
- Lists – Used to create lists, combo boxes and dropdown combo boxes that contain text or graphics
- Menus – Used to create floating menus, which can also include other buttons
- Edit controls – Used to create single and multi-line edit controls
- Status controls – Used to create status and progress bar controls
- Tree controls – Used to create tree-like controls similar to those found in the Windows Explorer application

The Application Foundation Class user interface controls make it easy to mix and match different types of controls. It's very easy to mix text and graphics on many of the controls. Next, I review the more common user interface controls. I will show you how to create static controls, buttons and

menus using the Application Foundation Classes. Note: For your reference, in the last Visual J++ Corner column I provided a complete hierarchy of the Application Foundation Classes, including all of the user interface classes.

Static Controls

There are three primary types of static controls in the Application Foundation Classes. They are the static text control, the static graphic control and the static item control (used to display both text and a graphic image). The static text control is represented by the `UIText` class, the static graphic control by the `UIGraphic` class and the static item control by the `UIItem` class. The `UIText` constructor takes a single parameter – the text to be displayed. The `UIGraphic` control also takes a single constructor parameter – an `Image` object which represents the image to be displayed. Finally, the `UIItem` control takes three important parameters – the text to display, an `Image` object representing the graphic to display and text alignment constants. Listing 1 pre-

sents a simple panel class that creates each of these static controls. Figure 1 illustrates how these controls are rendered.

Buttons

There are three primary types of button controls in the Application Foundation Classes. They are push button controls, radio button controls and check box controls. The push button control is represented by the `UIPushButton` class, the radio button control by the `UIRadioButton` class and the check box control by the `UICheckButton` class. The `UIPushButton` constructor takes two parameters – the text of the button and style of the button. You can create three button styles: using the `UIPushButton.RAISED` constant, you can create a standard push button; using the `UIPushButton.TOGGGLE` constant, you can create a two-state button; using the `UIPushButton.TRITOGGLE`, you can create a three-state button. If you substitute a `UIGraphic` or `UIItem` object for the text parameter in the constructor, you can easily create a button that displays a graphic or text and a graphic. The `UIRadioButton` and `UICheckButton` constructors each take a single parameter – simply the text to display on the button. Listing 2 shows you how to create buttons using the Application Foundation Classes. Figure 2 illustrates how the buttons defined in Listing 2 are rendered.

Menus

Finally, the last Application Foundation



Figure 1: How the static controls in Listing 1 are rendered



Figure 2: How the button controls in Listing 2 are rendered

StingRay

full

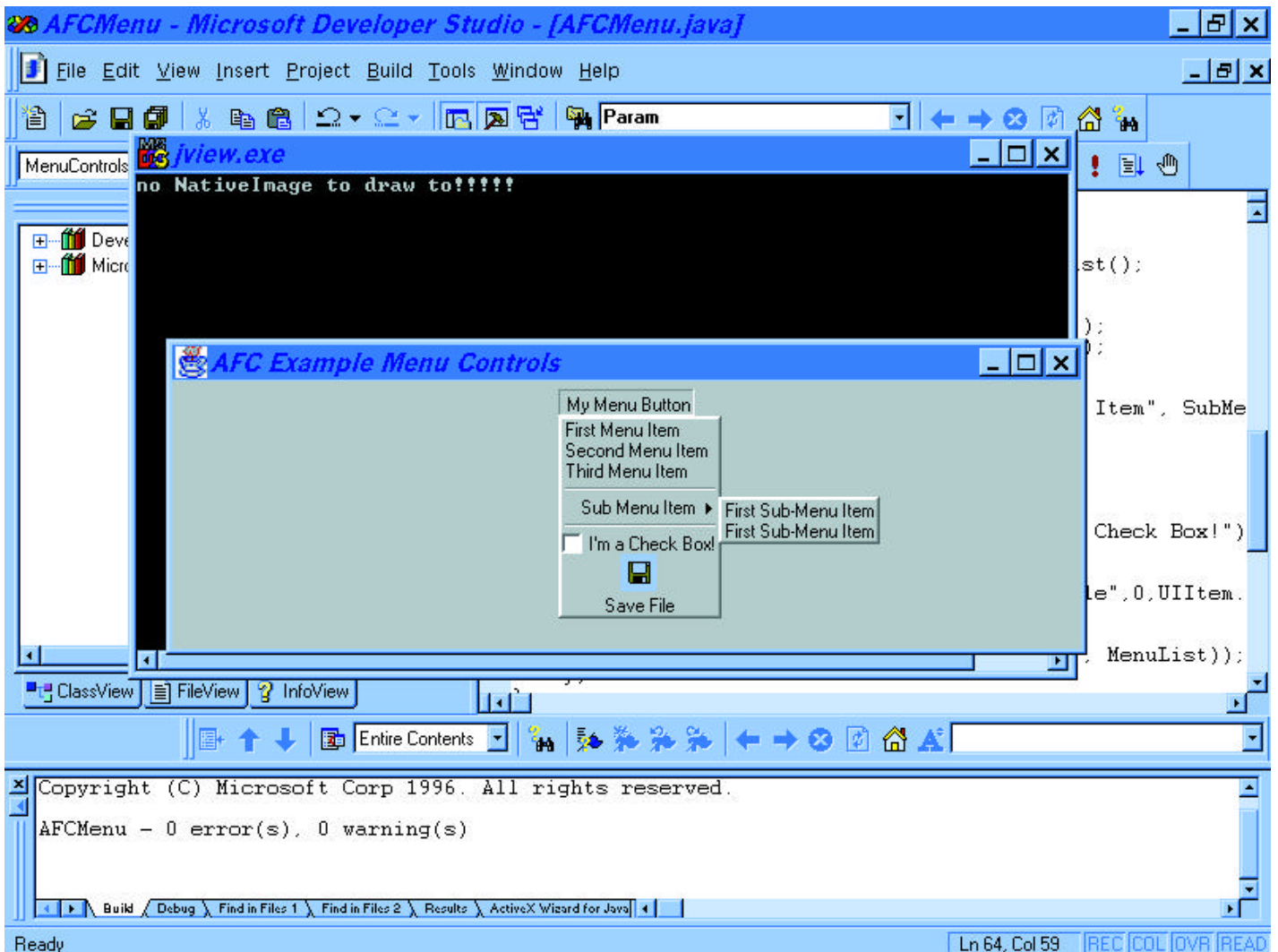


Figure 3: How the menu control in Listing 3 is rendered

Class I will review here is the Menu control. Menu controls are comprised of three main classes: `UIMenuItem`, which represents an actual menu item; `UIMenuList`, which represents a list of menu items; and `UIMenuButton`, which represents a menu push button. When a menu push button is pressed, the menu is displayed. Menus are pretty easy to put together. As outlined in Listing 3, you begin by creating a menu list. The `UIMenuList.add` method is called to add new menu items. You can add text strings to create simple text-based menu items or you can create complex menu items by adding different types. In the example in Listing 3, I have added a check box menu item by simply adding a `UICheckBox` object to the menu list and a graphic and text menu item by adding a `UIItem` object to the menu list. It's also easy to create sub-menus: just add another menu list to the main menu list. Once you have constructed your menu hierarchy, the `UIMenuButton` item is created, which marries a menu list with the menu button. Figure 3 illustrates the menu defined in Listing 3.

J/Direct and the Future of AFC

The Application Foundation Classes are certainly powerful. I've only touched the tip of the iceberg; the Application Foundation Classes also include a set of Enterprise Classes. The Application Foundation Enterprise Classes include the ability to build transaction-enabled applications in Java that interface with Microsoft Transaction Server and also provide access to legacy information systems and databases.

Microsoft has announced their J/Direct initiative, which will be part of the Java Virtual Machine in Internet Explorer Version 4.0. J/Direct should strike fear into the hearts of the pure Java camp, as it will allow Java applications written for the new Microsoft Java Virtual Machine to directly access the Windows API. Does this mean that developers will rush to create applications using the J/Direct interfaces? I prefer to wait and see. Even though this industry moves at faster than light speeds, sometimes it takes the future a lot longer to get here than you'd expect. It is my opinion though that some sort of marriage between

the Application Foundation Classes and J/Direct is somewhere in the near future, with a set of Application Foundation Classes optimized for the Windows VM that utilize J/Direct and another portable set that doesn't. This could create a win-win situation for developers, since they could still develop portable applications, but rely on the speed and close integration with the Windows platform when running on the Java Virtual Machine. Will I be right? I don't know, but it's an interesting enough story to keep me tuned in every day. ☘

Next Month: The Visual J++ Database Wizard: Is The Magic Right For You?

About the Author

John W. Fronckowiak is the author of "Teach Yourself Database Programming With Visual J++ In 21 Days" and "Teach Yourself OLE DB and ADO In 21 Days", Sams.net Publishing. Is there a special aspect of Visual J++ you would like to see discussed? Send your ideas and comments to John at john@buffnet.net.



john@buffnet.net

Kovisky full

Listing 1: Creating static controls using the AFC.

```
// Create a panel that contains AFC static controls

class StaticControls extends UIPanel {
    public StaticControls() {
        Toolkit tools = getToolkit();

        // Load Images
        Image disk = tools.getImage("disk.gif");
        Image arrow = tools.getImage("arrow.gif");

        setBackground(Color.lightGray);

        // Static Text Example
        add(new UIText("Example"));

        // Static Graphic Example
        add(new UIGraphic(arrow));

        // Static Item Example
        add(new UIItem(disk, "Save File", 0,
            UIItem.ABOVE));
    }
};
```

Listing 2: Creating button controls using the AFC.

```
// Create a panel that contains AFC button controls

class ButtonControls extends UIPanel {
    private UIRadioButton RadioButton;
    private UICheckBox CheckButton;
    private UIPushButton PushButton;
    private UIPushButton ImagePushButton;

    public ButtonControls() {
        setBackground(Color.lightGray);
        Toolkit tools = getToolkit();

        // Load Images
        Image disk = tools.getImage("disk.gif");

        // Radio Button Example
        add(RadioButton = new
            UIRadioButton("I'm a Radio Button!"));

        // Check Box Example
        add(CheckButton = new
            UICheckBox("I'm a Check Box!"));

        // Push Button Example
        add(PushButton = new
            UIPushButton("I'm a Push Button!",
                UIPushButton.RAISED));

        // Image Push Button Example
        UIItem item = new UIItem(disk, "Save File",
            0, UIItem.ABOVE);
        add(ImagePushButton = new UIPushButton(item,
            UIPushButton.RAISED));
    }
};
```

Listing 3: Creating button controls with a check box.

```
// Create a panel that contains AFC a menu control
class MenuControls extends UIPanel {
    public MenuControls() {
        setBackground(Color.lightGray);
        Toolkit tools = getToolkit();

        // Load Image
        Image disk = tools.getImage("disk.gif");

        // Create A Menu List
        UIManager MenuList = new UIManager();

        // Add Standard Menu Items
        MenuList.add("First Menu Item");
        MenuList.add("Second Menu Item");
        MenuList.add("Third Menu Item");

        // Add A Menu Break
        MenuList.add(new UILINE());

        // Create A Sub Menu Item
        UIManager SubMenuList = new UIManager();

        // Add Sub Menu Items
        SubMenuList.add("First Sub-Menu Item");
        SubMenuList.add("First Sub-Menu Item");

        // Add The Sub Menu To The Main Menu
        MenuList.add(new UIMenuItem("Sub Menu Item",
            SubMenuList));

        // Add A Menu Break
        MenuList.add(new UILINE());

        // Add A Check Box Menu Item
        MenuList.add(new
            UICheckBox("I'm a Check Box!"));

        // Add an Image Menu Item
        MenuList.add(new UIItem(disk, "Save File",
            0, UIItem.ABOVE));

        // Create The Menu Button
        add(new UIMenuButton("My Menu Button", MenuList));
    }
};
```



Greenbrier full

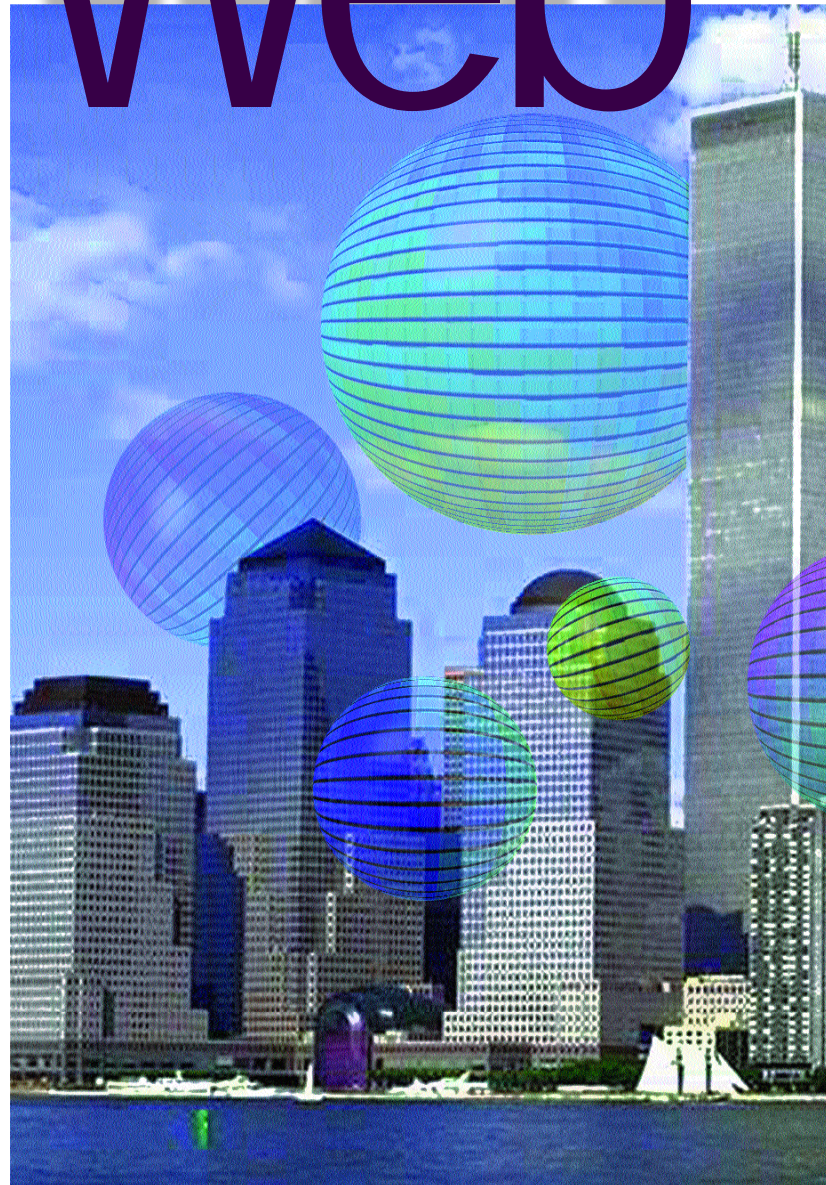
Pushing Open the Warehouse with the Web

Moving crucial information
onto the desktop with
control and flexibility

by Judy Rawls

Since the inception of the Internet, there has yet to be a topic as controversial as "Push". In fact, it's hard to pick up a trade journal that doesn't talk about it. So, without writing a dissertation on the value of "Push", let's explore an extension of this idea. "Managed Push" is a methodology that lets everyone in the information chain have the tools, access and security to obtain enterprise-wide data when, where and how they need it.

For example, you're the CFO. You walk into the office 20 minutes before a board meeting. You flip open your laptop, click on your Web browser and, voilà: The report you need on operating expenses by branch is sitting there waiting for you. No scrambling to pull spreadsheets together, no scratch pads with columns of arithmetic, no last minute conference calls to finalize the numbers. You're done, on your way, ready to go. At the same time, throughout the corporation, other decision-makers, knowledge workers or analysts, call them what you will, have exactly the information they need in time for their meetings.



This scenario is only a recent possibility, a result of combining "Push", the warehouse and the Web.

How Many Ways To "Push" ?

"Push" can be defined in three broad categories: indiscriminate, selective and publish and subscribe. Indiscriminate "Push" is an ongoing trickle of updated information, used to distribute data from inside, or outside, the organization on a real-time basis. Examples include news releases, stock quotes and sports scores; distributed to the desktop, pager or fax. For single or small groups of subscribers, this is a very effective method for up-to-the-minute information. The downside occurs when network bandwidth is clogged with information that may not be critical to a company's success.

Selective "Push" gives the user, and the company, more control. Instead of a constant channel of information, subscriptions can be defined at the server level. A filter sorts and prioritizes data to be delivered to the desktop, funneling external and internal information to subscribers. This is effective for pushing pre-defined sets of information to classes of employees, say by job title. These smaller packets of information place less stress on network traffic. The drawback is profile administration. Management tools are not yet available for large companies and the server administration alone often requires several people.

Publish and subscribe is a balance of subscriber control and administrator process management. Instead of pushing massive amounts of information to the desktop, data is published one

time to the server by the administrator. Subscribers then determine when they want the information, and under what criteria. When the scheduled time arrives, the information is pushed to the users. Subscription is usually by time or event, requiring scheduling and monitoring software. Users have the ability to discontinue a subscription or change the delivery schedule. Administrators need not worry that resources are being used to push information to a user who has changed their requirements. To achieve this flexibility, subscription interfaces and management tools must all be in place.

Separating Informational from Operational Systems

Today, there are two sides to information systems: operational, or transaction, and informational. The operational side includes the systems familiar to most people. These are the HR, payroll, finance or manufacturing systems that run the daily operations. Operational systems require data to be stored in a specific manner to achieve maximum performance from those systems processing the day-to-day tasks of running the business.

The informational side was created to pull corporate information together. Informational systems support analysis and decision-making and require the data to be stored in an entirely different manner for correctness and clarity. Data warehousing lies within the informational, or decision support, side of the organization's information systems and has its own unique requirements for information flow.

Data from multiple operational systems is collected, transformed and integrated onto a single platform for decision support. These warehouses are today referred to as "data marts", "operational data stores", "reporting databases" or "query databases". All have the same function – to provide a universal resource for knowledge workers to locate information upon which to make their daily decisions.

Pushing Data to the Warehouse

Pushing data requires that there be several elements in place. First, there must be a process to gather data from legacy, operational and external databases and push it to the warehouse. Often, data is in different media, formats or types and can have multiple definitions. Moving data to the warehouse is not simply a matter of writing a program, but of intelligently delivering the desired data to the appropriate targets.

A second consideration is the sheer volume of data being moved. Operational databases may contain gigabytes of data from systems having continuous update cycles. Specifically, updating the warehouse with only the desired data, which may be just the changes since the last update (known as Changed Data Capture), is an excellent opportunity to use indiscriminate "Push". Timing and data selection are paramount to "pushing" data from the operational side to the warehouse.

Pushing data from operational systems through the warehouse and out to the desktop is a process management opportunity. To effectively manage and distribute corporate data, a set of tools must be integrated into the data movement/push scenario. These tools include built-in scheduling mechanisms and recovery and restart, to ensure job completion and data integrity.

Central to process management is a transformation engine, or concurrent manager. This engine can avoid bottlenecks and single points of failure while executing processes in a distributed manner. In addition, by load balancing work across multiple servers the concurrent manager creates a scalable solution that IT administrators can use to leverage data across multiple environments. Finally, system tuning and process refinement is simply achieved with statistics on rows moved, transformed and loaded; in effect, creating a continuous information loop.



The result is a tightly integrated “managed push” solution which gives corporations the tools and processes to actively leverage the wealth of information available from enterprise data. The data warehouse becomes a reservoir of mission-critical information to be shared across the organization. By combining scheduling, monitoring, movement and recovery facilities, organizations can automate the information

environment and directly contribute to the success of any warehouse project.

Managed Push Opens the Warehouse on the Web

OK, so now you have a data warehouse with data you’ve “pushed” from the operational systems. Nevertheless, there’s one essential issue: a data warehouse is a technical solution to a business problem.

Decision makers, who would most benefit from the warehouse, are the least likely to use a query tool or run an SQL-based report writer to get information. The IT teams that built the warehouse have a wealth of integrated, corporate-wide information but do not have a way to deliver it to information consumers. Fortunately, the Web has reversed that scenario.

A recent study published by Robertson,

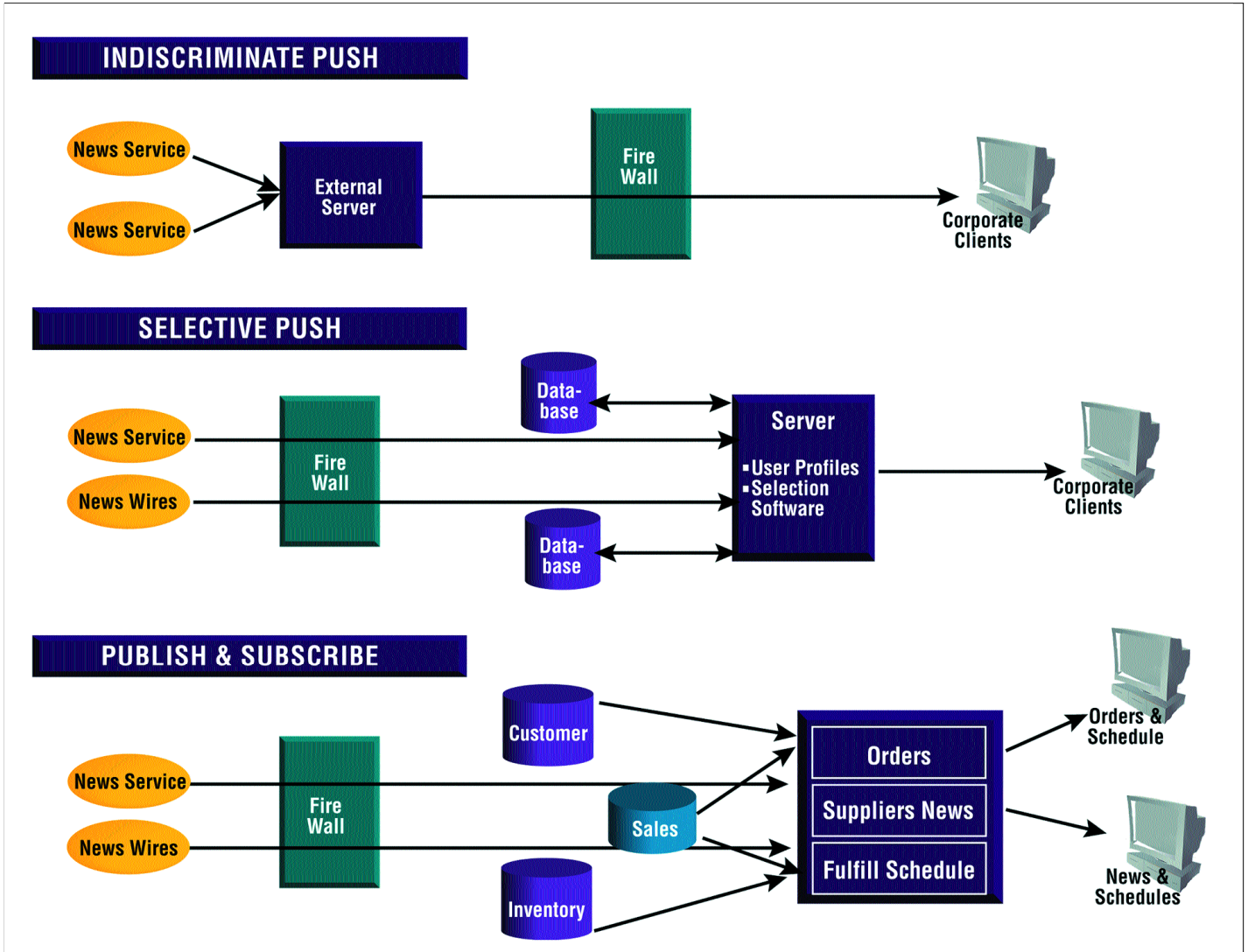


Figure 1: Will “Push” Pan Out?

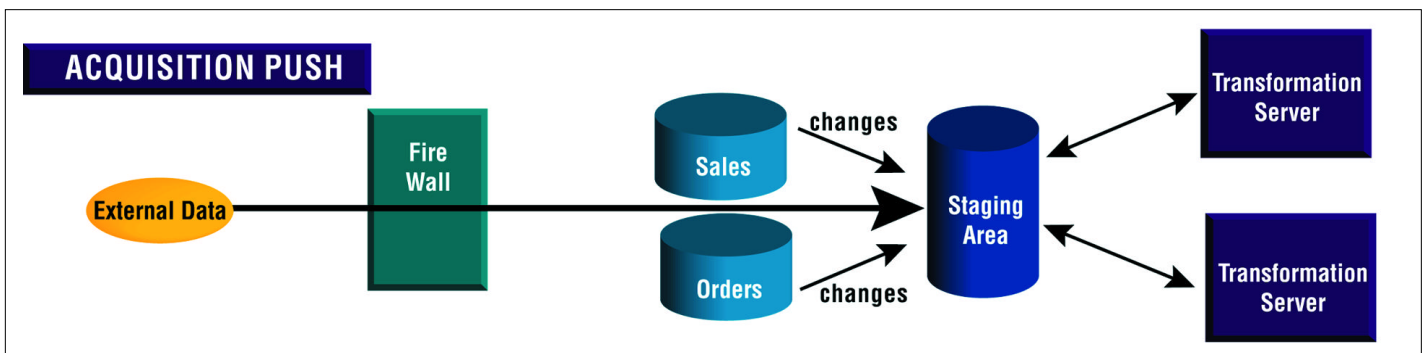


Figure 2

IONA

full

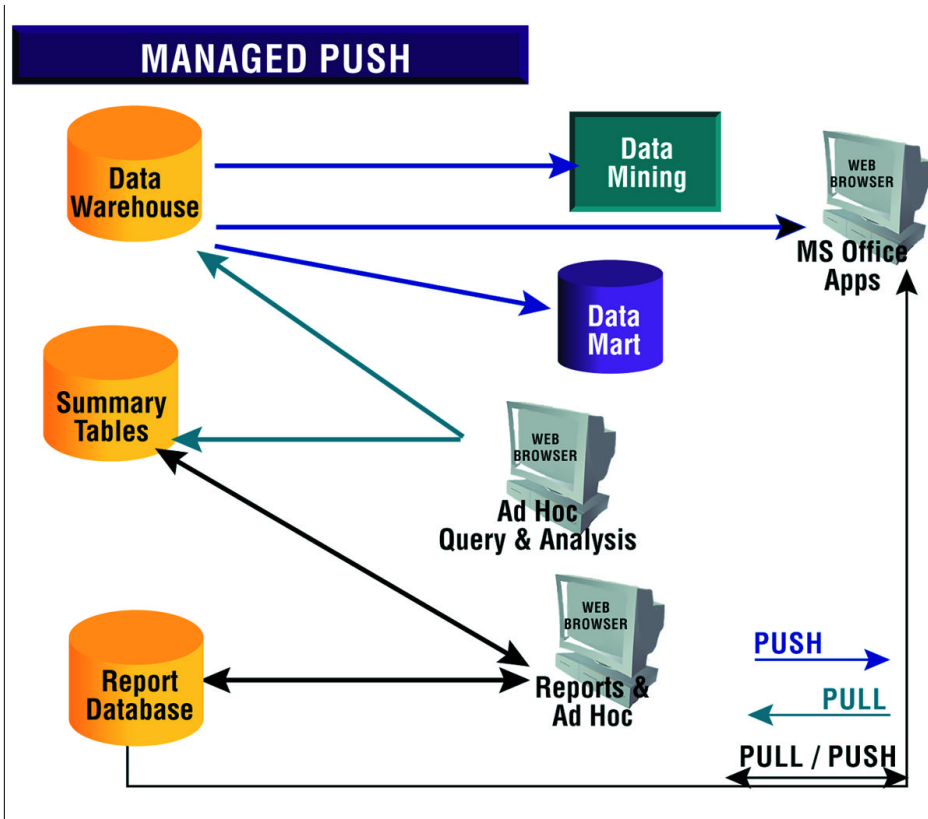


Figure 3

Stephens & Company noted that they expect the Internet to replace client-oriented tools due to the low cost of deployment, easier administration and distribution enabled by the Internet. (Robertson, Stephens & Company, August, 1997, "Turning Data Into Decisions and Information Into Insights," p.13.) It is not a stretch of the imagination to see how the Web will open a whole new venue for accessing, reviewing and sharing organizational data. The challenge is defining the process and having the tools to enable such an application.

With an informational system infrastructure in place, the most effective method to push information to the knowledge workers is "publish and subscribe." This "Push" method extends "managed push" to the desktop by giving knowledge workers the power to schedule only the information they need, when they want it. These knowledge workers may require a stream of defined reports and information in a repeatable and consistent manner. "Publish and subscribe" is a perfect application to deliver integrated, enterprise-wide information to the desktop, without burdening the recipient with additional training or systems knowledge.

Additionally, "publish and subscribe" can streamline the information delivery process. It is much more efficient to publish views of the data and assign security

to those views once, in the warehouse. Users may subscribe to views or subsets of views, and determine the delivery format and schedule. By publishing data once, to the server, the data is contained in a central location, reducing data redundancy. All

users are now able to obtain and reference the same information, providing a common basis for decision-making across the organization.

Pushing From Beginning to End

"Managed Push" moves the crucial business information out of operational systems to the desktop, with maximum control and flexibility. The universal interface, cost-effective deployment and accessibility of the Web browser have transformed the information in the warehouse into a resource available to any knowledge worker within the corporation.

Only when these leading technologies, combined with the push methodology, are used together, can organizations bridge the gap between knowledge workers and operational systems. The incorporation of a "managed push" ensures that companies can leverage their data in a productive, measurable and secure manner. ●

About the Author

Judy Rawls is Vice President of Marketing at D2K. Prior to joining D2K, Rawls was Director of Product Planning for Prism Solutions. Before that she worked with Bachman Information Systems as technical marketing consultant. Judy holds a Bachelor of Arts degree in Mathematics and History from Southern Methodist University, Dallas and a Master of Business Administration from the University of Utah. She can be reached at jrawls@d2k.com



jrawls@d2k.com

Pushing the Envelope with Java for "Managed Push"

One company that is using the Web and "Push" to provide a data warehousing solution is D2K, Inc. in San Jose, CA. Rather than settle for the limitations of HTML or the inefficiencies of CGI, D2K has taken a 100% Java approach to Tapestry's user interfaces. Each user interface is a fully self-contained, pure Java applet. Using pure Java applets ensures that Tapestry's interfaces will work on any platform and keeps D2K out of the Microsoft vs. UNIX vs. Macintosh wars.

Tapestry by D2K uses a Java server to manage the Java applets and interact with its many databases. To open the warehouse with the Web, D2K chose 100% Pure Java on the server to simplify porting between the many different server platforms accessed in an enterprise-wide deployment.

Tapestry's object-relational catalog provides optimum performance for a central repository of information about each data source, system users and data maps detailing how data is mapped from one data source to another. Java Servers use JDBC connections to the catalog to retrieve instructions for how, and when, data should be moved.

Java also plays a role in data movement. Native method calls from the Java server are made to the source database to extract the data. Today, native method calls are the most efficient way to extract data from DBMSs. In the near future, Remote Method Invocation (RMI) promises better response time, portability and integration with Java.

Besides moving data between databases, Tapestry also creates and delivers information to end-user desktops. Current delivery formats include Microsoft Excel, text, HTML and JavaCharts. Many of Tapestry's users prefer to receive JavaCharts for the added convenience that upon arrival users can manipulate the chart, changing formatting and titles. Tapestry's charting capability stands out in a crowded marketplace because, unlike most tools today, Tapestry doesn't use any plug-ins or HTML.

As for the future, D2K is getting ready to announce their second release of Tapestry, which will include a development workbench for data warehouse design. Written entirely in Java, this interface will employ drag and drop drawing controls to lay out data movement and database structure. D2K's continuing use of 100% Pure Java will ensure that Tapestry is available on the widest range of platforms and that its users benefit from upcoming enhancements in Java.

Team Share

1/2 Ad

Digital Think

1/2 Ad

Pushing Out to your Clients



An Interview with
Mark Bowles,
Technical Guru of
TIBCO

Java-in-the-browser makes any
reasonable Web browser a potential
TIB sender or listener...

by Edward Zebrowski

Since Push technology is making such a big splash with the Internet community, naturally we wanted to take a closer look at the emerging technology. *Java Developer's Journal* recently had a chance to interview Mark Bowles (bowles@tibco.com) of TIBCO. TIBCO recently released the latest version of its TIB/Rendezvous Push product.

TIB/Rendezvous is an industrial-strength messaging tool which allows application developers to build scalable distributed applications that share data across LANs and WANs. These applications run on heterogeneous platforms and communicate transparently with self-describing data messages and subject-based addressing.

Mark is the Technical Guru (his official title) of TIBCO and was very enthusiastic about doing the interview.

JDJ: How did you get started with Push technology?

MB: TIBCO started life in 1985 (80 Net years ago) as a large-scale custom systems integration company. We got a major contract with a large New York investment broker to build a digital trading system for them. At the time, most traders (the guys you have come to know and love from *Liars' Poker* and *Bonfire of the Vanities*) had video monitors delivering up-to-date stock prices to them. In their business, timeliness and accuracy of current stock prices are everything – they trade for their own accounts on such thin margins that a mistake of a quarter or an eighth can make the difference between making and losing millions of dollars.

The problems with the video systems were several: the data was presented in a form which was convenient for the sender, not the traders; it could not be used in spreadsheet and other modeling calculations since it was just a video image; and reconfiguring a trading floor often meant hundreds of thousands of dollars in electrician charges to rewire the video cables under the floor. What was needed was a system which transmitted all of the data through a local area network and which transmitted it in usable form – floating-point numbers tagged by ticker symbols, for example. We pioneered this form of trading system in 1986-87. This, in short, is the ultimate Push application.

Today, this system is in use in a majority of the large investment banks and mutual funds around the world. During active trading hours (which is most of the time in this global world), trades and quotes stream into a pair of fault-tolerant servers in the computer room of the bank; they are converted to TIB records and published on the network based on their ticker symbol or CUSIP code. Recipients have an application on their desks which knows how to subscribe on the same basis. Since the underlying transport is IP multicast, each message is sent only once regardless of how many people are listening.

JDJ: Why Push technology?

MB: Glad you asked. Push – publish/subscribe, as we call it – is fundamentally different from request/reply and both paradigms are important in today's high-speed, high-throughput, just-in-time world. With request/reply, a client asks a question and a server answers it. The server is the repository of knowledge (often a database) and the client is seeking to extract one or more facts from it. Ask another question, get another answer. What's missing from request/reply, however, is proactive notification of change. Say user "A" changes the value for the number of red widgets in the

finished goods inventory. The database dutifully stores the new value and might even run some triggers within itself in order to make sure it is self-consistent. But what if users "B" through "W" would dearly like to know when the value changes? Neither the database, nor the request/reply paradigm, provides any help with this. To solve their problem, users "B" through "W" have to poll the database periodically, asking a question like, "Well, how many red widgets are there now?". This generates a huge amount of useless database load and net traffic and is, in a fundamental way, missing the point. The original transaction changed the world in some way; the system should take the responsibility to proactively propagate the news of that change to people who want it. That's why our flavor of push is called publish/subscribe. Push is proactive notification of change. It's distribution of the news in the most general sense of the definition of the word "news."

JDJ: Where do you see the technology going?

MB: Eventually, it will be as all-pervasive as TCP/IP is today – that is, Push will be built into operating systems, routers, switches, firewalls. While this will take some time – probably several years – I know it will happen because of the popularity of the ad-hoc Push products which currently are on the market without the help of O/S's and so on.

JDJ: Which brings me to an obvious comparison. How does TIB/Rendezvous compare to Marimba's Castanet and other similar products?

MB: It comes down to three things:

1. "True Push" – TIB/Rendezvous ("RV") not only presents a publish/subscribe interface to its users, but it actually performs the dissemination of information in a true pro-active fashion. Messages are sent on the network only when changes actually happen, pro-actively, from the publisher (Marimba's transmitter) to the subscriber (tuner). We use IP multicast to accomplish this and to get the huge one-to-many fan-out required for real, scalable Push applications. Most of the rest of the so-called Push applications, including Castanet, use TCP/IP – a point-to-point protocol – under the covers and are really repeatedly polling from the tuner to the transmitter. This is a little like covering a hole in the floor with carpeting – it doesn't help the problem, just makes it look better.

2. "End-User versus Middleware" – This is why most of these vendors (our marketing folks, kfarmer@tibco.com, can get you the current, complete list) have signed up to use RV under their covers. Most of these applications come from young, venture-funded companies, trying to add value to the Internet. Such companies cannot afford

to attempt to "change the world" as we can – they have to put products on the ground which work with today's operating systems, firewalls and so on, and if they are inefficient on the network side, well, that's a problem we'll solve after we're profitable. TIBCO, however, is a middleware vendor and we view these firms as our current and potential customers.

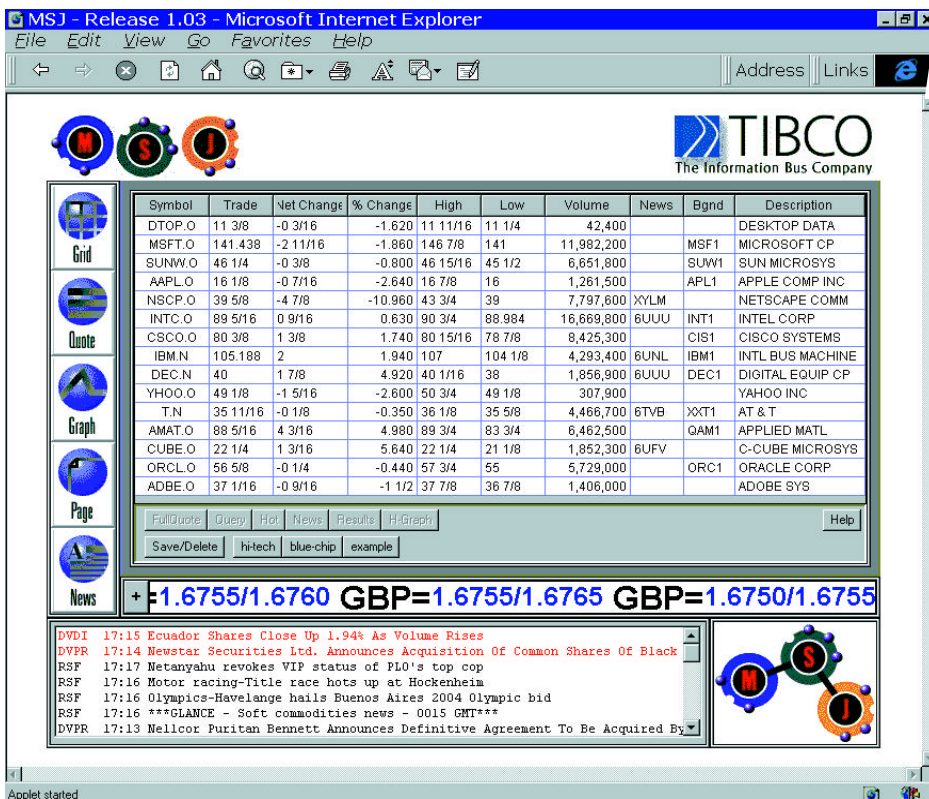
3. "Web-oriented" – Also, most of the other firms have focused on the more specific problem of "pushing" selected Web pages from Web servers to individual clients – that is, "Make sure I always have a current copy of the page www.playboy.com on my laptop so I can enjoy it on the airplane." This focus allows the companies to be more consumer-oriented, but it also leaves behind a much more compelling problem, in our opinion: "My firm needs to know

Push is proactive notification of change. It's distribution of the news in the most general sense of the definition of the word "news."

when a customer, collaborator or competitor changes its mind about something."

JDJ: That said, what are the benefits of TIB/Rendezvous?

MB: RV, being transmitted via multicast, uses much, much less network traffic than solutions which are built on top of TCP/IP or HTTP. Also, the use of multicast harnesses the compute power of the routers and switches of the network, so that the sender does not have to do a quantum of work for each (receiver, message) pair. Non-multicast solutions must do an explicit "send" for each receiver for each message; the more the sender is successful (i.e., the more listeners it has), the higher its compute load – the good news is that we have a hundred thousand listeners; the bad news is that we have to buy a Cray and lease two



Two views of real-time quotes and news "pushed" to TIBCO's browser-based application, MarketSheet for Java.



more T3s to accommodate them.

JDJ: Mark, who's using TIB/RV today?

MB: Our financial applications now run in most of the large investment banks around the world. The technology is also very useful in running manufacturing operations

and, in particular, semiconductor fabs. Four of the five largest semiconductor manufacturers are using TIB/RV or its predecessors in controlling all of their new fabs. The most successful stock-quote Web page out there right now is Yahoo!'s; it is powered behind the scenes by TIB.

JDJ: where do you see the technology five years from now? Ten?

MB: I've touched on this one earlier. Five or ten years is an incredibly long time in Net years! I'm quite sure that Internet multicast and the other basic technologies underneath TIB/RV will be installed and running in the fabric of the world's compute network, including the grand Internet, in five years.

JDJ: Is this technology for large corporations or can small businesses use it as well?

MB: We sell TIB/RV through our Web page and have dozens of small customers as well as our larger banking, manufacturing and other clients. Our marketing and sales focus is on the Global 500, but we also encourage small OEMs who use RV to build custom applications.

JDJ: How does TIB/Rendezvous fit into a typical corporate Web?

MB: Corporate environments, whether they are Web-based or not, are the target environment of RV. Java-in-the-browser makes any reasonable Web browser a potential RV sender or listener, with the corporate network handling the transport via IP multicast. Think of the number of things which change, day-to-day, hour-to-hour, second-by-second, in a corporate environment. With TIB in your Web page, you can sit back and watch them change on an otherwise static Web page, rather than repeatedly hit the refresh button in order to see what's new.

JDJ: What do you think of Java? Where does it fit in? ActiveX?

MB: On the surface, Java and AX are frameworks through which we make RV happen; we offer interfaces to both, as well as RV versions for virtually all hardware/OS platforms. Personally, I am also very bullish on Java; we started actively tracking it a few months before the first Java-One and introduced our first Java-based products there.

JDJ: We want to thank you, Mark, for this very interesting discussion. Readers who wish to read more about TIB/RV, or want to download a free trial version, should surf over to <http://www.tibco.com>. The TIBCO Web site features FAQs, white papers and additional information about TIB/RV and other TIBCO products.

About the Author

Edward Zebrowski is a technical writer based in the Orlando, Florida area. Ed runs his own Web development company, Zebra Web, and can be reached on the Net at zebra@rock-n-roll.com.

 zebra@rock-n-roll.com

DCI Full Ad



Push for Corporate Intranets

Fertile ground for networking applications that make a difference

by Tony Davis

Many people consider Push an application specifically for the Internet. In fact, there is a growing number of corporations using Push as an Intranet application at the departmental level as well as corporation-wide. The Forrester Group predicts that by the year 2000, the enterprise market for Push will be approaching \$300 million.

Why are Fortune 1000 companies turning to Push? Because it is much more than delivering CNN or Wired news to desktops. Serious Push is the proactive delivery of business-critical data to people who need to know and act. Knowledge workers are inundated with information, much of which is irrelevant to their work efforts. Yet among the mountains of data in e-mail, databases, the Internet and voicemail is information that could make them more efficient, productive and smarter in the execution of their work. Serious Push enables the right people to receive the personalized data that is critical to their work in real time: notification of low inventory levels, new client wins or losses, sales over a certain threshold for a product or in a given area or automated competition tracking. Normally, users must look for this kind of data; Push enables its delivery to wherever they are working. Push technology enables the right data to find the right people at the right time.

And why is Java critical to application development in the Push market? Pushed data must be able to come from any back end source and reach any client or application on any platform. Obviously, Java and technologies like CORBA are particularly suited to development in an environment that requires support for multiple sources and platforms. The Push architecture you work with should be open on both the client and server sides. In BackWeb's case, Java developers can design applications that automatically receive live data via BackWeb channels. The data can then be

manipulated and sent back upstream to a database or to another recipient. The data could be coming from databases via JDBC and the users identified via JNDI. Writing to Java APIs on the back end and pushing data to Java-based applications on the client side enables portability and scalability.

Using Java in the Push environment is also well suited to building strategies for rapid application assembly using components or applets. Charting, spreadsheet or OLAP functionalities can all be written as beans which can be snapped together and reused depending on the requirements of a given Push application. They may be used interchangeably to receive data through channels and present it in the most advantageous way. A component could also be used which allows users to prioritize where or how they want to be notified of critical data.

One of the primary advantages of Push in an Intranet environment is that it is a low cost way to ensure that software applications throughout the enterprise are up-to-date, including application code and Java applets or templates and graphic elements. For example, a database tracks the Java components that a given user needs to run a sales territory monitoring application. You don't want the user to have to wait for those components to download when the application is launched in a browser. Using Push technology, the applet components can be delivered automatically to the user before use so that they launch locally. If you are pushing software upgrades over your Intranet, it is critical that you can easily target specific groups of users. BackWeb, for example, provides a sophisticated, rules-based targeting system that allows software upgrades to be narrowcast to a specific group of users or even personalized for a single user. The targeting system could also be extended through a 'hook' to a database using JDBC or any other connectivity method, allowing personalization for unique users, as in our sales territory tracking example.

The ability to do differential downloads

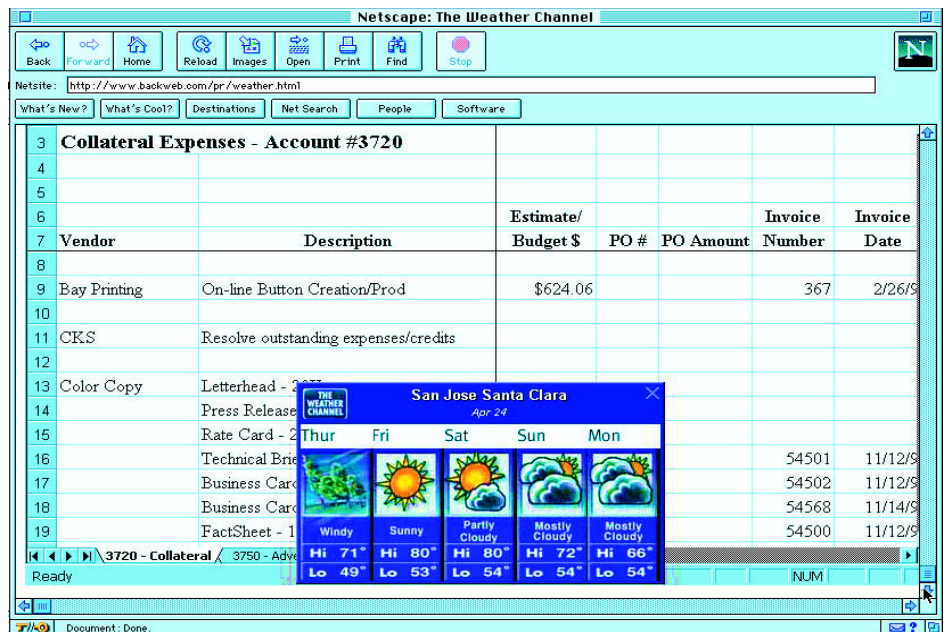


Figure 1: Users can receive InfoFlashes like this one for the Weather Channel, on their screen without disturbing or interrupting the application they are working in. The BackWeb-based Weather channel delivers customized Weather to users.

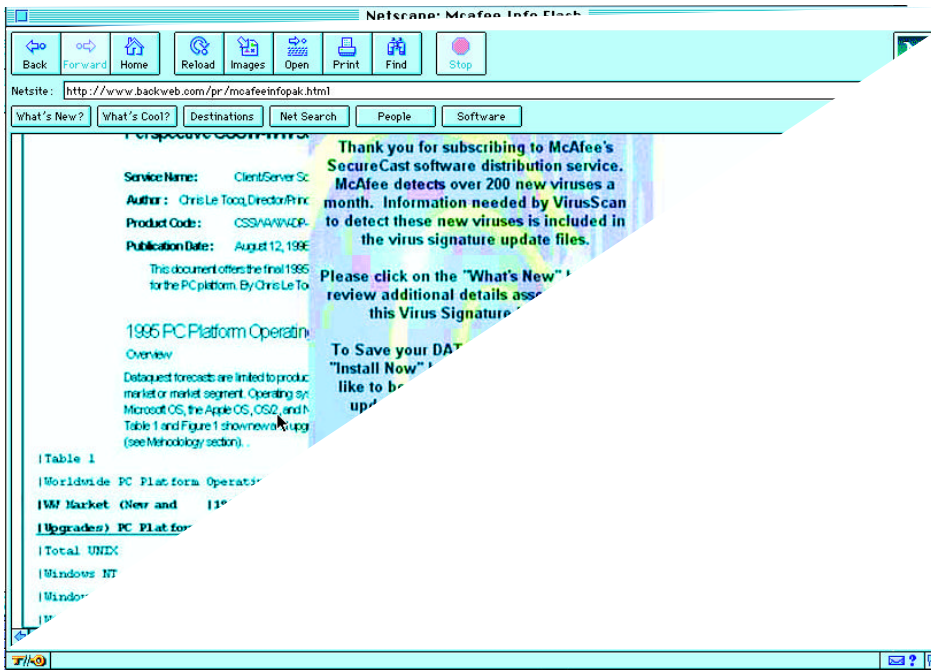


Figure 2: McAfee delivers anti-virus updates in the background through BacWeb's software delivery feature.

is a critical factor for implementing Intranet Push to deliver software and updates. Frequently, only a few class files in the application actually change, which is especially true of Java. It is much more efficient (and bandwidth-friendly) to push only the changed portion instead of the whole application. In addition, compressing the pushed content can save about 50 percent of the available bandwidth.

Since application distribution involves pushing a relatively large amount of data, it is very important that it has minimal impact on users' systems – it can't interrupt their work. And dial-up users need a way to complete downloads over multiple sessions without starting over each time. BackWeb calls this issue "politeness" and provides its Polite Agent technology for background, incremental downloading during a user's idle on-line time.

Scalability is another critical issue in implementing Intranet Push. If you have a one megabyte application and you want to push it to 100,000 users, you need a smart solution that will know to use multicasting to push the same content to many users, versus a "naive" Push solution that will move that one megabyte application 100,000 times – which would equal 100 gigabytes of data going over your network.

A well-designed Push infrastructure can be used to effectively deliver applications and a lot more. Polite Push is an ideal way to deliver rich, multimedia content. The user gets both high quality and immediate gratification since the delivery of the information is separated from its presentation – it is not displayed until fully downloaded.

It is evident that there are numerous technology benchmarks that separate seri-

ous Push solutions from the news delivery and Webcrawling technologies that gave birth to this market. The hype surrounding Push is beginning to subside and powerful applications are being built. Data will come from anywhere and must be able to reach any client, so Java is a perfect development environment for this exploding market. Push is fertile ground for Java developers seeking to build networked applications that make a difference in the way information moves through businesses. ●

About the Author

Tony Davis is Senior Vice President of Product Development at BackWeb Technologies, a leading provider of enterprise Push solutions. He was previously founder and CEO of Push software provider Lanacom, which was acquired by BackWeb in July 1997. Davis was the architect of Delrina's WinFax family of products. His pioneering concept of low-cost, PC-based fax communications established WinFax as the worldwide leader in fax communications software and established Davis as a premier developer of user-friendly interfaces and tools. He can be reached at tdavis@lanacom.com



tdavis@lanacom.com

Criteria for Selecting a Push Solution

- Does it deliver any kind of content including Java applications, applets or beans, executables, multimedia, any file format?
- Can it pull data from any source, including enterprise databases? Or is it limited to HTML content?
- Is it network and Internet polite, so that you remain in control of your company's bandwidth and resources?
- Is the Push technology massively scalable?
- Does it have proactive alerts, letting you notify your employees that information has arrived in the application they are using at the time?
- Does it offer anti-spoofing and other critical forms of security?
- Can you do software upgrades and asset management automatically?
- Does it take full advantage of and integrate well with both Microsoft and Netscape browsers?
- Does it allow you to do differential updates and automatic compression of large files?
- Does it let you determine to whom the content will be sent – by person, function, geography or by type of hardware or plug-in?
- Does it let you manage the content delivery in any function or does delivery require IS control?
- Does it let you deliver different priorities of information intelligently?
- Does it let you control the timing of when content is sent, when played and how often played?
- Can you choose between scheduled and real-time delivery protocols?
- Is it bandwidth savvy, making large file downloads practical, even dial-up connections?
- Does it give you total branding and creative control?
- Is the technology open enough to allow you to build your own applications on top using Java or any other language?
- Does it let you integrate high-value external news channels?



Making Java More Secure PART 2

Two ways to control potentially misbehaving applets

by Jay Heiser

A New Buzzword

The Java security community has begun to use a new buzz phrase, 'mobile code', to describe Web executable content like Java, JavaScript and ActiveX. The name is meant to distinguish it from non-Web forms of executable content, such as Microsoft Word macros and PostScript. All executable content has the potential to cause security problems – MS Word macro viruses have caused more damage than all other executable content attacks combined. Some future operating environment, such as Project Spin, may well be robust enough to resist multiple forms of executable content attack, but for the time being, the security controls of each type must be handled separately. This month's article will examine two commercial efforts to centrally control mobile code; specifically, misbehaved Java applets.

Finjan

Finjan became the first commercial developer of a Java security enhancement last fall when they introduced SurfinShield, a software add-on that allowed workstation users to prevent the execution of Java applets by their browsers. Evidencing a high rate of new product introduction, Finjan hasn't limited itself to Java and the current product, SurfinShield Xtra 2.0, guards against both undesired Java applets and ActiveX code. SurfinShield, which claims to prevent e-mail attacks, has the potential to protect against hostile code or URLs to hostile code that are anonymously sent to a Java-enabled e-mail client.

SurfinGate is an organization-level control designed to prevent damage from hostile mobile code. Positioned as a supplementary firewall, it checks Java applets for hostile intent before they ever reach the client. Running on a server at the organiza-

tional boundary – normally immediately behind a firewall – it allows an administrator to set an organizational Java security policy for allowable Java capabilities. Applets that attempt to write to a file or communicate with a host other than the originator are identified as such by Finjan's scanner, allowing a decision to be made as to whether the applet should be provided to the requesting user.

Finjan's product maintains a database storing identification and security-relevant information on every applet it encounters. It actually scans the byte code to analyze the applet's intentions, storing its characteristics in the database. While it will be able to take full advantage of signed applets, it can also provide some of the characteristics of signing even to unsigned applets. Every incoming applet is identified and only those meeting the security criteria are allowed inside the organization.

SurfinGate compares incoming applets to a list of known applets in its database, including all known hostile applets (a fairly small list today). Unrecognized applets are scanned for potential hostile behavior, generating an Applet Security Profile that is stored for the next encounter with the applet. The ASP, whether previously stored or just generated, is compared to the site and user security policies to determine if the applet is acceptable for use.

A free download and lengthier descriptions are available at <http://www.finjan.com>.

Digitivity

A subsidiary of APM Ltd., an eight-year-old British consulting firm, Digitivity has managed to generate much media attention for its approach to Java security. Digitivity is unique in not running the Java applet on the workstation at all, but instead running it

on a special server and sending the graphical output to the workstation using a proprietary protocol.

Digitivity's product sprang from a consulting project that APM was performing for a large bank. They were developing a Java applet for use by the bank's customers. Since the bank had already disabled their own access to Java applets on the Internet at their firewall, their customers asked why they should be expected to use the bank's Java applet over the Web when the bank itself was unwilling to do so. Good question. As APM began developing a solution that could allow the bank's partners to access the Java applet without putting them at risk from other Internet Java applets, it became clear that a solution to a universal problem might have wider commercial applicability.

The Applet Management System currently consists of two server components and a client component. The AppRouter, which normally would sit directly behind the organizational firewall, is used as the Web proxy by the client browser. It intercepts requests to download Java applets and instead of providing the applets themselves to the browser, it forces browsers to load Digitivity's proxy applet. The AppRouter provides direction to a CageServer, usually located outside of the firewall, to download the Java code. Following instructions from the AppRouter, the proxy applet connects through the firewall to the CageServer and uses the BrowserBridge protocol to display the graphical interface of the applet, which actually runs on the CageServer. BrowserBridge protocol is unique to Digitivity but is functionally identical to Citrix's WinFrame or the X Window System.

The architecture, which is unique among the others examined in this series, offers advantages in scalability. There is no need for a one-to-one correspondence between AppRouters and CageServers. A single AppRouter can serve multiple CageServers. Beyond what the sandbox provides at the browser level, Digitivity touts a number of benefits for this architecture:

- Quarantines outside or unacceptable mobile code so that it never runs on the user's desktop
- Centralizes Internet applet execution to

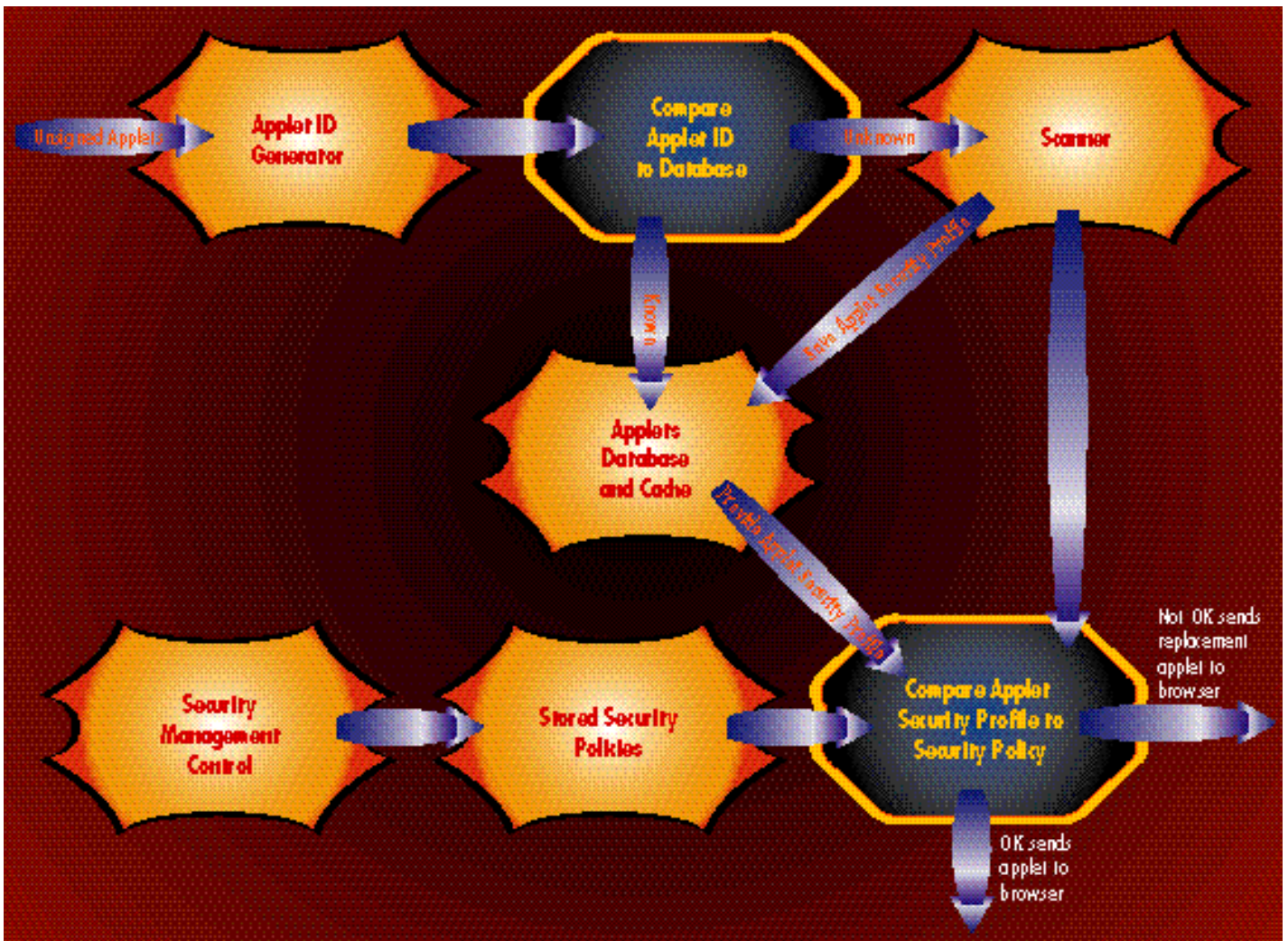


Figure 1: Finjan identifies and signs unsigned applets.

- facilitate common management
- Prevents users from making undesirable changes in their environment that violate organizational security policies
- Runs applets on a platform optimized as a secure environment

Digitivity plans several important upgrades over the next year. The Policy Cage will take advantage of signed Java applets to allow centralized management of access control. The Enterprise Cage will support the deployment and coordination of multiple CageServers and will interface with the major network management systems.

According to Andi Bruno, Digitivity's Director of Marketing, "People resonate with the very physical form of protection we offer and the way in which our three announced products provide a series of stepping stones from applet security to applet management to enterprise applications. Once established, it is centrally controlled and administered. The entire process is transparent to end-user. We've taken the locus of control away from the desktop/browser and end-user and put it

under central administrator control and management at the server level."

For more information on Digitivity, see www.digitivity.com.

Is All This Necessary?

This is a good opportunity to apply some of the security concepts that we've discussed over the last few months. Finjan and Digitivity, along with the students and faculty at the Secure Internet Project and Project Kimera, make security countermeasures. The decision-making process leading up to the application of a security countermeasure is called risk analysis. To analyze the risk of allowing users to run Internet-borne Java applets, we'll examine vulnerabilities, threats and the cost of a successful attack.

Vulnerability

At this point in time, the current versions of Internet Explorer and Netscape Navigator seem resistant to known attack methods. The process of examining the products for weaknesses that might be exploitable is called flaw hypothesis methodology. This approach has identified

virtually all of the security bugs discovered in the popular browser products (HotJava included). Attempting to predict the future based on past events has been compared to driving a car by looking out the back window. Point taken, but if your car doesn't have a front window, you can make some educated guesses based on what you see out the back window. Sometimes the past record is all we have. Historically, computer science has identified a steady stream of browser security holes that the browser vendor has quickly plugged with updated code. Looking out the back window of the browser lifecycle car, it's reasonable to assume that Netscape and Internet Explorer probably have unknown security holes.

Threat

So far, the threat population of hidden hostile Java code is empty. There are no public records of known attacks through the use of Java applets (or ActiveX). Well-marked Web pages containing hostile code are available to the curious, but there are no documented incidents of this code being used for hostile attack. People who are known to attack computers – either

Digitivity Java proxy applet on browser interfaces with running Internet applet on CageServer

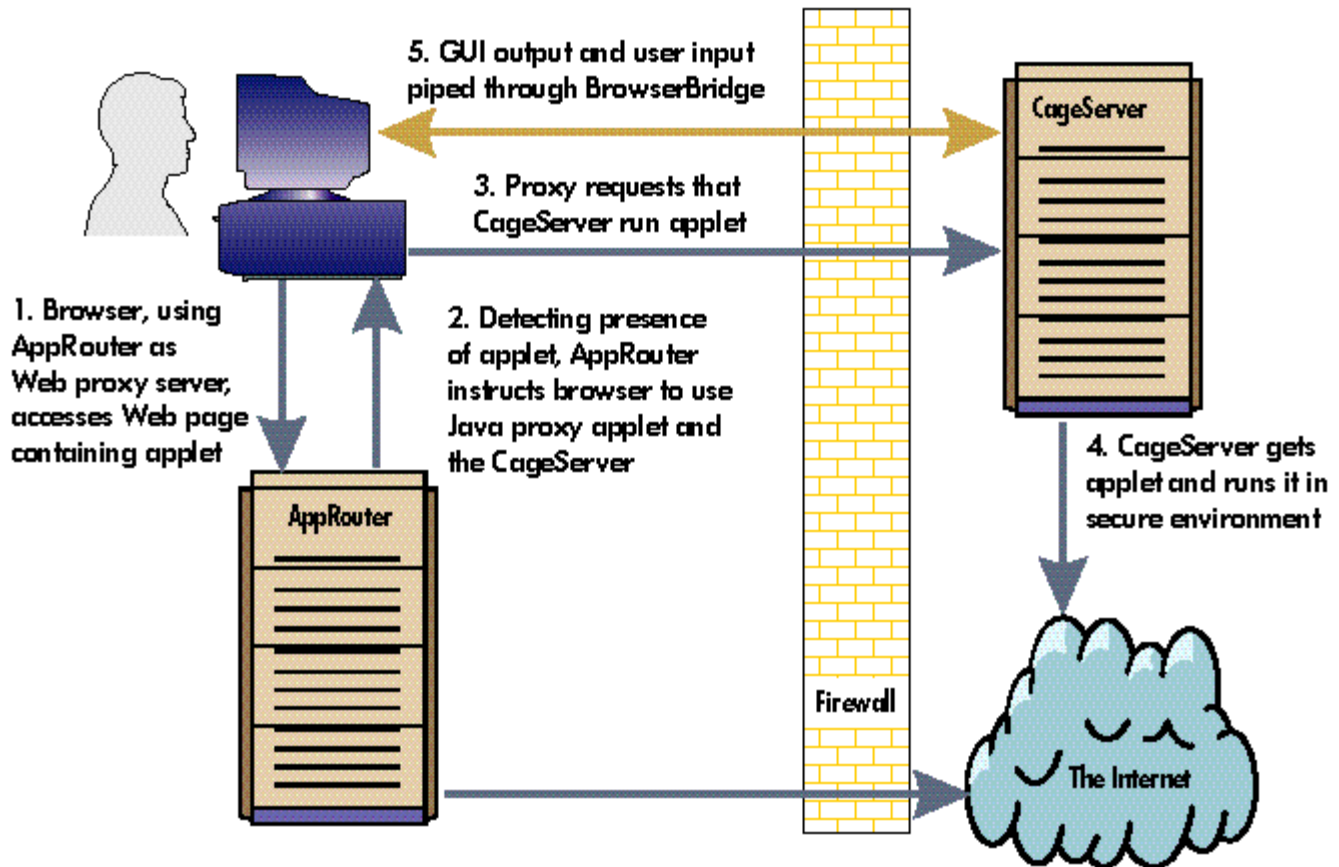


Figure 2: Digitivity Java proxy applet on browser interfaces with running Internet applet on CageServer

through viruses, Trojan horses or by directly cracking a system over the Internet – represent a statistically significant threat population. Based both on history and knowledge of human behavior, it is reasonable to assume that easily exploitable vulnerabilities will be subject to attack. So far, Java applets have not become a convenient means of system exploit, unlike viruses, which are anonymous and self-replicating. Hostile applets are hard to insert and they don't spread, making them much less efficient for system compromise.

Value

What would it cost your organization to be hit with annoyance applets or data theft applets? You've probably encountered computer viruses and maybe Trojan horses transmitted by e-mail. What was the effect on your organization?

The Bottom Line

The vulnerability of individual systems is low, but unrecognized weaknesses are likely. The current threat is almost nonexistent, but a significant population of poten-

tial attackers exists who would take advantage of Java vulnerabilities if it were convenient to do so. The cost of a successful exploitation varies from organization to organization, but could be incredibly high for organizations handling financial transactions or other sensitive data. Analysis of risk, choice of policy and implementation of security countermeasures are judgment calls that will legitimately vary from organization to organization.

While there are still no documented cases of hostile mobile code being anonymously transmitted through a Web page, there are many cases of hostile code, especially macro viruses and Trojan horses, being sent through e-mail. If you are actively receiving Internet mail, but not protecting against e-mail threats, you are vulnerable to a well documented threat. Consider protecting your organization from e-mail pathogens before spending resources on Java threat countermeasures. If you or your management has strong reservations about the use of external Java code, but must use it to remain competitive, one of

these products will go a long way towards reducing your concerns and vulnerability. My personal opinion is that for most organizations, the threat represented by Java applets on the Internet is minimal. However, as Java becomes increasingly common and signed applets gain system privileges, centralized and effective control mechanisms – especially for external code – will become routine. Next month's article will look at Netscape's and Microsoft's plans for providing this centralized Java policy control. ●

About the Author

Jay Heiser is the Director of Internet Products for HomeCom Internet Security Services, where he is currently providing network security consulting to several major financial institutions and retail chains. He has lectured on information security in the US and Europe at events such as InfoWarCon, The Internet Conference and FOSE. He can be reached at jay@homecom.com.



jay@homecom.com

Web Commander

by Luckman Interactive

Designed to help you through your task with a one-stop shopping, soup to nuts solution!

by Vince Barnes



I want... I want... I want... Seems like those childlike wants of ours never disappear, they just change slightly to match our changing environments! If you have to build a Web site, whether for internal use on an Intranet or for publication on the Internet, you want a complete set of tools. You want a Web Server program with a simple administration interface, a Browser, a Web Page Editor (or two!), a graphics program with image map capabilities and some extras for accessing databases from your site, providing search engine capabilities and supplying fill-in forms for your site's visitors to use. Perhaps you want to provide secure pages for conducting credit card business on the Web and want tools to simplify that task. But, you don't want to spend a fortune to get them.

Enter Luckman Interactive with Web Commander. This Web package for Windows 95 or Windows NT comes with just such a set of tools and is designed to ease you through with a one-stop shopping, soup to nuts solution! For this review, I tested Web Commander on a 686 120 MHz machine running Microsoft Windows NT 4.0.

The package comes with a well-written book of documentation containing easy to follow descriptions of every process from installing TCP/IP and getting connected to the Net to setting up your server(s) and building your pages.

Installation is easy -- a standard setup process. At the end of the installation, Setup presents the opportunity to run the Web Site configuration. With a simple six-tab dialog box, all the basic features of a server are defined. This includes a description for the site, its IP address and port, its root document directory, the Webmaster's address, virtual paths (aliases), mime types (of

which all you are likely to need have been included by default) and directory browsing features. Also included are access permission rules, logging requirements and styles.

The administration tab allows you to specify users and groups of users, then allow or deny access to specific directories by users or groups. You can also grant or deny access to the site by specific IP addresses or domains (using wildcards if needed). This is also where you enable and configure logging. The extended logging capability includes referral source, client browser and other useful information beyond what is included in the basic log file. The extended log file is in a comma delimited format, most convenient for importing into a spreadsheet program for analysis.

Styles use a wizard to define a style to be applied to all documents in a specific directory and its subdirectories. The style wizard lets you specify a header that can include a graphic, a background color or wallpaper, and a footer that can include a graphic, descriptive text and an e-mail address. This

Web Commander

Luckman Interactive

Phone: 800-711-2676 or 213-614-0966

Web: <http://www.luckman.com>

Email: info@luckman.com

Price: \$129 for Windows 95, \$249 for Windows NT

Platforms supported: Windows 95, Windows NT

provides an easy method to ensure a consistent look for a set of pages.

When the server has been defined, its settings are saved in a configuration file (.KWB). The click of a button starts the process of creating a new server. You could create a separate server for "live" and "development" sites or, if you are providing Web services, one for each domain name. This configuration file provides a very useful side benefit; it can be moved, along with its directory structure, from one machine to another as a complete, intact, Web site. How's that for a development/backup/management aid?

Once configured and running, the site is managed through the Web Commander Window (see Figure 1). This is used to start, pause or stop servers (sites) and monitor a variety of useful statistics such as hit rates, referrals, active connections, errors and general server statistics. There are three

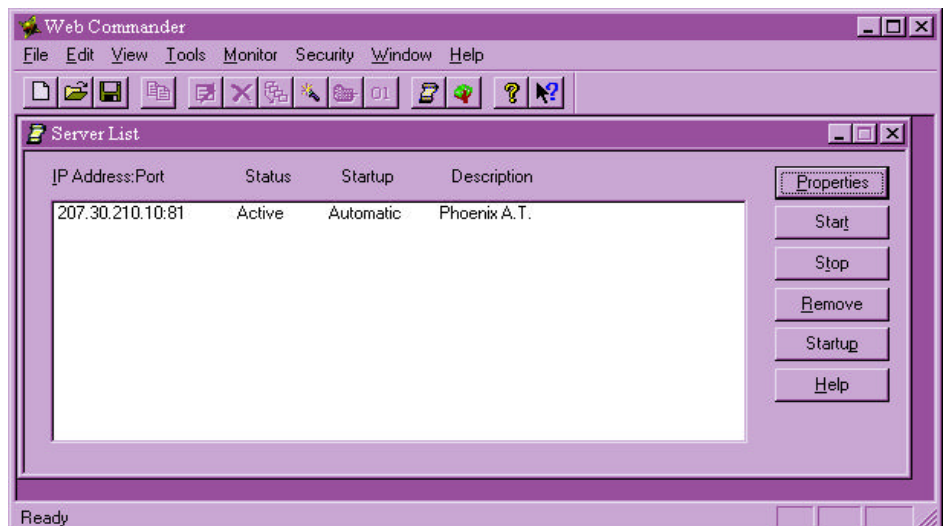


Figure 1: The Web Commander Window

other very useful items tucked away in here.

First, under Tools/Options we find "Install Service". Almost glossed over in the book and not found by me at first, is the answer to a very important question for those of you running NT. Yes, it does run as a service!

Also under Tools are Show Tree and Tree Tools. Show Tree opens up a window with a tree structure view of the Web site for this server, together with file and server information. Branches (subdirectories) are expanded and collapsed in standard windows fashion. Documents containing hyperlinks and graphics can also be expanded to reveal the files and destinations included. Selecting an item and clicking on "Tree Tools" (or right mouse clicking) lets you open it in your favorite browser editor. The choice of each is specified in the Tree Tools /Helpers dialog.

The third item is the Security menu. Here there are wizards that greatly simplify the process of building a secure Web Server. The first wizard walks you through the steps of obtaining a security certificate; the second installs it when it is received from the issuing authority (Verisign). By adding a server on port 443 (the SSL default) and using these two wizards, you can correctly set up a secure server easily and quickly.

Web Commander also includes a wizard for creating Web pages. This is certainly a fast and easy way to create pages! It has the ability to create a corporate, personal, order form, feedback, database or even a generic Web page. In each case, it asks for basic information pertinent to the desired page, together with general questions about background colors, hyperlinks, headers and footers, and finally asks for a location in which to save the page. It is not hard to create a reasonable looking set of pages in just a few minutes with this wizard.

As an alternative to the Web Page Wizard, or to further customize the pages created by the wizard, there is the Web Studio. This is a full-featured Web editor (also known as the InContext Spider). The program has a "logical" section on the left for placing elements such as "Heading" or "Paragraph" on the page, and a "content" section on the right for entering the content of the element. There are good examples supplied with the package that help you to quickly understand how the program works. It is easy enough to use and includes a good selection of HTML mark-ups. Personally, I would like the option of seeing the resulting HTML code in full while I am working on a page. This program also appears to be a 16-bit program. The file open and save dialogs don't have long file name capabilities and wouldn't let me Save As an .HTML file! This program is an adequate starter

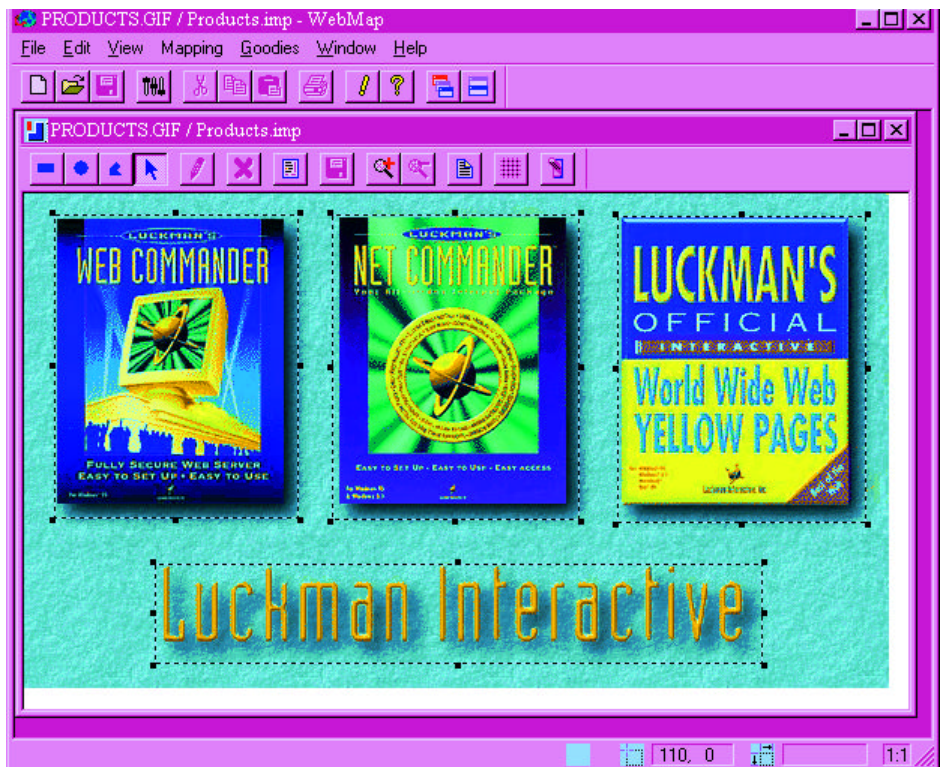


Figure 2: The WebMap program window

tool, but I wouldn't rely on it for long.

The Web Map program allows for the creation of image maps based on .GIF image files. Rectangles, ovals and polygons are supported and the creation of an image map is simple. Web Map includes a useful grid function that helps to keep hot areas nicely aligned. There is also a zoom for fine detail work along with cut and paste editing.

The Web commander package also includes Excite for Web Servers and Wide Area Information Server (WAIS) tools. These tools enable powerful search capabilities for your site. Their use is well documented in the book, including step-by-step instructions. If your site is going to contain a lot of pages these search engines help to make your clients' lives a lot easier.

The book includes excellent documentation on the use of Server Side Includes (SSI), Common Gateway Interface (CGI) and the new Information Server Application Program Interface (ISAPI). There is also an appendix with some very good examples of CGI and PERL scripts, including a guest book, a questionnaire form, a bulletin board and a variety of very handy little routines.

WebCharge is an application that allows credit card transactions over the Internet. It works by calling out on a modem to obtain an authorization code from the clearing house. Configuration is simple enough but only allows for one of four clearing houses. There is no way to add details for your clearing house if you don't use one of these four (NaBANCO, MDI, VISA, TDSI). If you are in luck, however, then you can also use the

manual entry screen, which allows you to enter transactions that don't originate on the Web.

In conclusion, I feel Luckman's Web Commander is very good. The Web Server itself is definitely the strongest aspect of the package. I was very impressed with its ease of use, flexibility and comprehensive feature set. The wizards come next as an excellent aid to accomplishing otherwise complicated tasks quickly and easily. There is room for improvement in the Web Page authoring area, although the included program gets the basic job done. The image mapping tool is simple to use and fills a much needed slot in the tool box. There is a lack of a graphical development and manipulation tool in the kit; a void which I would definitely like to see filled. The documentation is excellent; it is readable, informative and makes a good reference. Overall, while not exactly a "one-stop shopping" solution, it is close to it and represents a very good value; an excellent server and a good starter tool kit. ●

About the Author

Vince Barnes is the General Manager of Phoenix Applied Technology, Inc., an Internet Service Provider and computer services company in Kissimmee, Florida. His career spans more than twenty years of computing, from software engineering to network design. Though now primarily involved in management and strategy, he likes to continue in very much of a "hands on" role. He can be reached at vince@phoenixat.com.



vince@phoenixat.com



Simple Thread Synchronization Tricks

Using thread-safe classes provided in the Java 1.0 core API to achieve thread synchronization without messy coding

by Brian Maso

One thing I really love about Java is the built-in support for multi-threading and thread synchronization. This multi-threading support is built straight into the Java Virtual Machine specification and it's built right into the Java language. Where in C++ you would have to use an OS-specific API to create and synchronize threads, the Java language provides the "synchronized" keyword for methods (as well as arbitrary blocks of code).

There are a few very common multi-threading synchronization problems that lend themselves well to being solved without resorting to creating synchronized methods or code blocks at all. Instead, you can borrow the thread-safe properties of the Java core API classes to perform your synchronization for you. Borrowing the synchronized code of already written core classes has three distinct advantages:

1. First, you don't have to perform the potentially tricky task of debugging your own thread-safe code. I once heard (and if the originator of what I'm about to say reads this, I'm sorry for stealing your clever saying) that trying to write multi-threaded code is like trying to keep track of a swarm of bees in your head – each bee individually. This technique avoids this difficulty.
2. The next advantage is a corollary of the first. When using core API classes, you don't have to worry about debugging since the classes have already been written and tested for you. Even more so than class libraries you might buy from a third party, the core classes have been used and re-used by hundreds of thousands of Java developers. This heavy use has pretty much ironed out all the kinks in the core classes.

3. The third advantage is that you can spend more of your development time on the domain of your problem, rather than on solving a multi-threaded synchronization problem that has probably been solved hundreds or thousands of times in the past. Using the thread-safe classes I will describe in this column might not give you the most optimized code. However, you can use the thread-safe classes to get your code up and running and working successfully before attempting to optimize it.

The three basic, recurring multi-threading problems I'm going to describe here I affectionately named "pipeline", "hungry puppies" and "waiting for the operator". Using the thread-safe classes of the Java core API, I've devised simple but effective solutions to these three basic problems, with the advantage that I have to write hardly any synchronized code. Well, maybe one or two methods, but still the solutions are cookbook. This means you can pretty much steal the code and use it without modification.

The simplest multi-threading common pattern is the "pipeline" pattern. In this pattern, a single thread is gathering data that is to be processed by a single other thread. The easiest version of this problem to solve using thread-safe core API classes is when the data source thread just pumps gathered data to the background thread in a non-synchronous manner; the background thread receives the gathered data and processes it as quickly as possible. If you set up a user interface so that the AWT event handling thread reads in user commands and passes them to another background thread for processing, then you would be following this pattern. In general, wherever the rate of

My previous articles have been about Java 1.0 issues, specifically ignoring Java 1.1. While the vast majority of Java Virtual Machines still follow the 1.0 standard, Java 1.1 will be in general release some time this year. It would behoove any Java developer to start looking at some of the features and modifications to the Java language between 1.0 and 1.1. This month's Tips and Tricks column covers a facility that is specific to Java 1.1. To run the example code, you must use a Java 1.1 VM, such as the one distributed with the latest Java Development Kit from Sun and JavaSoft.

incoming data should not be affected by the rate at which your program can process the data, you can set up a data pipeline to handle the situation.

Wherever feasible, I like to design my applications or applets to handle user input in this manner. Using the `PipedInputStream` and `PipedOutputStream` classes from the `java.io` package of the core Java API, I can synchronize the data gathering thread and the data handling thread without having to write a single line of synchronized code myself. If you are unfamiliar with these classes, you should consult documentation about them (in your favorite API reference book or in the on-line reference available at <http://www.javasoft.com>).

Any single `PipedInputStream` can be connected to a single `PipedOutputStream`. Since these two classes are available in the `java.io` package on any JVM, they're easy and painless to use. Byte data written to the `PipedInputStream` is buffered internally by the connected `PipedOutputStream`. In the pipeline pattern, the data gathering thread writes data to the `PipedInputStream`, while the data processing thread reads data from the connected `PipedOutputStream`. What's cool about the `PipedOutputStream` class is that its read method is a blocking call as long as no data is available to read. That is, the data processing thread will wait indefi-

nately during a call to `PipedOutputStream.read()` until data is written to the corresponding `PipedInputStream` by the data gathering thread. Listing 1 shows a simple version of this pattern to gather input and process it by two threads.

I call the next common multi-threading pattern “hungry puppies”. The name is meant to evoke the image of many processes vying for a constrained data source. This situation arises whenever you have multiple data processing threads which will all be receiving data originally gathered by a single data gathering thread. For example, recently I was a member of an engineering team developing a data-based application in Java. A single data gathering thread received what amounted to database queries. Up to five background data connections each had a thread associated with them that processed queries as they were gathered. In this way, a constrained number of database connections could be maintained and used in a round-robin style to process queries. Clearly, this design fits the description of a “hungry puppies” scenario.

The generic solution to a “hungry puppies” situation is given in Listing 2. Again, we exploit the core API’s `PipedInputStream` and `PipedOutputStream` classes to do our thread synchronization. Well, you end up having to write a small amount of synchronized code when using this technique, so you don’t get everything for free. But as I said, the solution is cookbook and will work without you having to think about it too much.

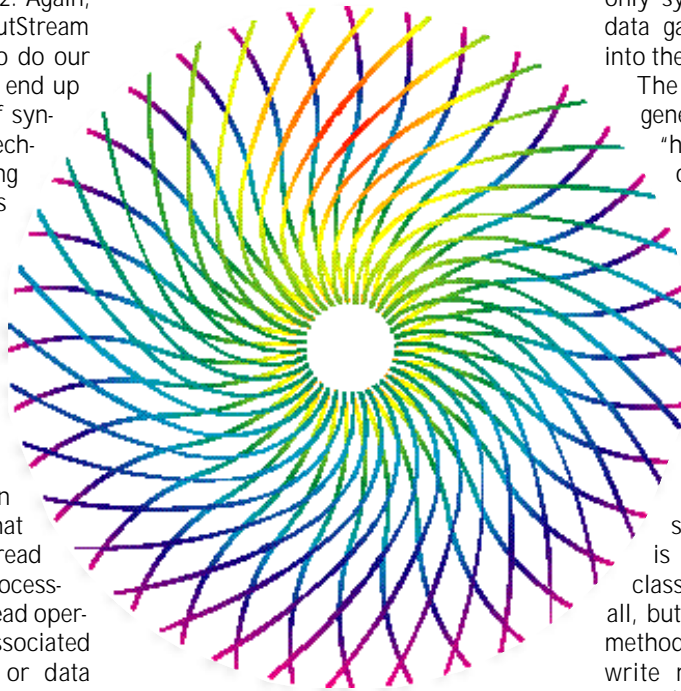
The architecture of the solution is to have the multiple data processing threads all read from the same `PipedOutputStream` object. The single data gathering thread writes data to a `PipedInputStream` object, just like I describe in the “pipeline” problem above. What we need, though, is an atomic read operation that each of the data processing threads will use. This atomic read operation will read all of the data associated with a single atomic command or data packet. For example, in my data-based application, I needed a way for the database connection threads to read an entire SQL query statement at one time.

As shown in Listing 2, the trick is to subclass the `PipedOutputStream` class and provide a synchronized “`readAtomicDataPacket`” method. The job of this method is to read all the data associated with a single data object written by the data gathering thread. Each data processing thread (the “hungry puppies”, so to speak) calls this method to receive data instead of using multiple calls to the

`PipedOutputStream.read` method. Since your custom “`readAtomicDataPacket`” method is synchronized, only one of the data processing threads will read from the `PipedOutputStream` at a time. The others will wait patiently until more data is available.

Follow this recipe whenever you encounter (or construct) a “hungry puppies” scenario and things will synchronize themselves automatically:

- Have the single data gathering thread write data to a single `PipedInputStream` using the stream’s normal write method.
- Subclass the `PipedOutputStream` class and provide a synchronized “`readAtomicDataPacket`” method whose job is to read all of the data from the `PipedOutputStream` associated with a single command or data packet. The reading of data can be through the inherited read method.
- Have all of the data processing threads call the `PipedOutputStream`-derived object’s “`readAtomicDataPacket`” method to receive all the data associated with a single operation.



The last common multi-threading situation I want to show you is a generic, simple solution for what I call “waiting for the teller”. It’s kind of the mirror image of the “hungry puppies” situation. In “waiting for the teller”, multiple data gathering threads are all gathering data to be processed by a single data processing thread. In this case, it’s usually not necessary for the data gathering threads to wait for gathered data to be processed before gathering additional data. Instead, each data gathering thread is

supposed to submit gathered commands or data packets to a single queue. The data processing thread will, in turn, read individual commands from the queue and process them as quickly as possible. The name of the scenario is meant to evoke the image of multiple people trying to get directory assistance on the telephone. Each caller must wait for the single directory assistance operator to handle them before going off and making other calls.

As you might have guessed, my generic solution to this problem (shown in Listing 3) requires that you subclass the `PipedInputStream` class to provide a synchronized “`writeAtomicDataPacket`” method. Each data gathering thread calls this method to submit a single atomic command or data packet to the queue of items to be processed by the single data processing thread.

The data gathering threads are not dependent on the data processing thread. That is, the rate at which data gathering threads operate is in no way dependent on when or how quickly the data processing thread is able to process data. Instead, the only synchronized activity is the various data gathering threads waiting for entry into the “`writeAtomicDataPacket`” method.

The astute reader can see that the generic scenario in which “pipeline”, “hungry puppies” and “waiting for the operator” all fall is a situation in which N data gathering threads are creating commands to be processed by M data processing threads. In the most generic case, both the `PipedInputStream` and `PipedOutputStream` classes would be subclassed and reciprocal “`readAtomicDataPacket`” and “`writeAtomicDataPacket`” methods would be added to the two classes. Just note that in the simplified case where either N or M is 1, then the corresponding stream class does not need to be subclassed at all, but instead can use the stream’s read method (if it’s a data processing thread) or write method (if it’s a data gathering thread). ●

About the Author

Brian Maso is a programming consultant working out of Portland, OR. He is the co-author of The Waite Group Press’s upcoming release, “The Java API SuperBible.” Before Java, he spent five years corralled in the MS Windows branch of programming, working for such notables as the Hearst Corp., First DataBank and Intel. Readers are encouraged to contact Brian via e-mail with any comments or questions at bmaso@europa.com.



bmaso@europa.com

Listing 1: Pipelines gather data from one thread and hand it to another.

```
//Java class of the data gathering thread
class DataGatherer implements Runnable {
    OutputStream os;
    public DataGatherer(OutputStream os) {
        this.os = os;
        Thread t = new Thread(this);
        t.start();
    }
    public void run() {
        while(!done) {
            //gather data command

            //write data command using successive calls
            // to os.write().
        }
        os.close();
    }
}
```

```
//Java class of data processing thread
class DataProcessor implements Runnable {
    InputStream is;
    public DataProcessor(InputStream is) {
        this.is = is;
        Thread t = new Thread(this);
        t.start();
    }
    public void run() {
        while(!done) {
            //read single data command using
            //is.read().

            //process command
        }
        is.close();
    }
}
```

```
//Main method that sets up the Pipeline.
public void MyMethod() {
    PipedOutputStream os = new PipedOutputStream();
    PipedInputStream is = new PipedInputStream();
    os.connect(is);
    new DataGatherer(os);
    new DataProcessor(is);
}
```

Listing 2: Hungry puppies has multiple data processors reading from a single data gatherer.

```
//Java class that gathers data is essentially
//the same as provided in Listing 1
```

```
//Java class that subclasses PipedInputStream
//to provide a "readAtomicDataPacket" method
public class AtomicInputStream extends
    PipedInputStream {
    public AtomicInputStream() {
        super();
    }
    public synchronized Object readAtomicDataPacket() {
        // read a single command or packet of data
        //potentially by using multiple calls to
        //the inherited PipedInputStream.read()
```

```
//method
    }
}

//Java class that processes data. Multiple
//objects of this type are created.
public class DataProcessor implements Runnable {
    AtomicInputStream ais;
    public DataProcessor(AtomicInputStream ais) {
        this.ais = ais;
        Thread t = new Thread(this);
        t.start();
    }
    public void run() {
        while(!done) {
            //Read each atomic command or
            //data packet from the AtomicInputStream
            //using its readAtomicDataPacket method

            //process each data packet individually
        }
    }
}
```

Listing 3: Multiple data gathering threads synchronize access to a PipedInputStream that acts as a queue.

```
//Data processing class is essentially the same as that
//provided in Listing 1
```

```
//AtomicOutputStream is a subclass of PipedOutputStream
//with a synchronized method to write a single atomic command
//or data packet.
```

```
public class AtomicOutputStream extends
    PipedOutputStream {
    public AtomicOutputStream() {
        super();
    }
    public synchronized void writeAtomicDataPacket(
        Object packet) {
        //write the data represented by the packet object
        //using multiple calls to the inherited write method
    }
}
```

```
//Data gathering class uses an AtomicOutputStream
//to synchronize writing atomic data packets or commands
// to a queue.
```

```
public class DataGatherer implements Runnable {
    AtomicOutputStream aos;
    public DataGatherer(AtomicOutputStream aos) {
        this.aos = aos;
        Thread t = new Thread(this);
        t.start();
    }
    public void run() {
        while(!done) {
            //Gather a single data packet or command

            //write the data packet or command to
            //the AtomicOutputStream using the
            // AtomicOutputStream.writeAtomicDataPacket
            //method.
        }
    }
}
```


Meckler Full Ad

Available NOW!

**DEVELOPER'S
JAVA JOURNAL**

Now our subscribers all around the world can read the latest issue of JDJ even before it goes to press! Check out this entire issue of JDJ at www.JavaDevelopersJournal.com in Acrobat/PDF format.

ADVERTISER INDEX

<u>Advertiser</u>		<u>Page</u>
Asymetrix www.asymetrix.com	800 448-6543	13
Bristol Technology www.bristol.com	203 438-6969	4
Critical Mass, Inc. www.j-v-m.com	617 354-6277	34
DCI www.DCIexpo.com/Internet	508 470-3880	31
Get Linked www.sys-con.com	914 735-1900	63
Greenbrier & Russel www.gr.com/java	800 453-0347	19
Java Internet Business Expo www.javaexpo.sbfforums.com	888 528-2397	35
JavaWorld www.javaworld.com	415 267-4527	65
KL Group Inc. www.klg.com	800 663-4723	68
Kovsky Conference Production, Inc. www.SoftwareSummit.com	800 481-3389	17
Marimba www.marimba.com/download	415 328-JAVA	39
Mecklermedia www.iworld.com	800 500-1959	45
Microsoft Corporation www.microsoft.com/VisualTools	800 621-7930	67
National Java Learning Center www.JavaLearningCenter.com	888 303-JAVA	48, 51
Object Management Group www.omg.org	508 820-4300	39
Rogue Wave Software, Inc www.roguewave.com	800 487-3217	11
Stingray Software Inc. www.stingsoft.com	800 924-4223	15
SunTest www.suntest.com	415 336-2005	6
Symantec cafe.symantec.com	800 453-1059 ext. 9NE5	3
WebLogic www.weblogic.com	415 659-2600	2



Project X

Creating an application without writing the code

by Alan Williamson

Last month, we started this column by looking at the development tool, Visual Café Pro. Visual Café Pro makes the perfect companion for the developer, giving instant access to the editor, on-line help and runtime environments. The main goal of this column will be to teach the reader the power behind Java by using Visual Café Pro, one of the most popular integrated development environments in use today.

There are many ways to teach Java. Some are good, some bad. I have always felt the best way to learn about anything is through example. Show somebody how it works and suddenly a lot of confusion is eliminated. For this reason, this column will develop a complete client/server application that can be deployed to both Internet and Intranet scenarios. Each installment will take the application a stage further, adding more and more functionality to it. Eventually, you will have successfully developed a full working application that may be deployed to a user base of thousands.

First, I will outline the application in question! I spent quite some time thinking about how I could demonstrate the workings of Java, but still come out with a piece of software that would actually be useful. As time went on, I rejected all of my ideas on the grounds that some were too big, some too complicated, too small, too easy. It looked bleak for a while. Then a colleague asked for an address for one of our clients, and I said, "It's in the company card file in the cupboard." A bolt of inspiration struck me. How many other people are in the same situation in an office environment; needing vital contact information held by only a handful of personnel? What is needed is a global contact database application.

OK, some of you may argue that this isn't exactly a revolutionary idea and, on

the face of it, you would be right. But in reality, how many companies actually deploy such a facility to their staff? Excluding external contact information, internal contact information is invaluable and very often overlooked. Applications such as Lotus Notes, Microsoft's Outlook and Pegasus Mail offer such functionality, but are very rarely used. Reasons for their non-deployment range from bulky client side front-ends to speed of use. What is required is something that is lightweight, easy to use, straightforward to install and operates on a global database.

Other benefits could include the ability to run on any platform and to be able to access it from outside of the office. Bit of a tall order, some might say, but look at what our requirement is. We want to have something that people can have instant access to, no matter what machine they are using, and it has to be small so that it doesn't irritate the user base: Java.

Project 'X' Specification

This series of columns will take you through the complete development of this

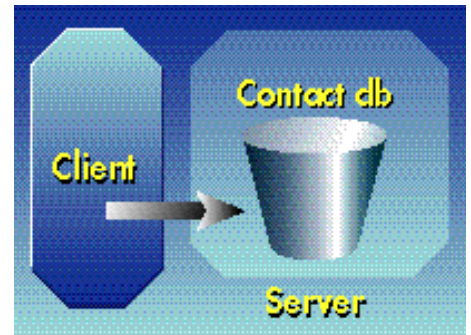


Figure 1

contact database, which we will name Informer. But before we open up Visual Café and start coding, let's outline the main features our new Informer will possess. First, it will be a Java applet. Since we will be using an applet, we are not restricted to the platform we are running from, as long as a Java-enabled browser exists in some shape or form. Informer will allow anybody to view the current list of contacts, add new entries or edit existing contacts. For our first version, we will give everyone the ability to add and change entries, which in reality isn't very practical.

Each applet will connect to the main database of contacts back at one central server, from which the applet will be downloaded, as shown in Figure 1.

From a database point of view, information stored about a particular person can be seen in Table 1. The database that this information will actually be held in will not matter too much. Since we will be using the dbAnywhere tool from Visual Café, the

Field Name	Field Description
NAME	Name of the contact
ADDRESS1	First line of the address
ADDRESS2	Second line of the address
COUNTY	County of address
COUNTRY	Country of contact
POSTCODE	Postal or ZIP code
WORK_TELEPHONE	Work telephone number
HOME_TELEPHONE	Home telephone number
E-MAIL	E-mail address
WEB_SITE	Web address

Table 1

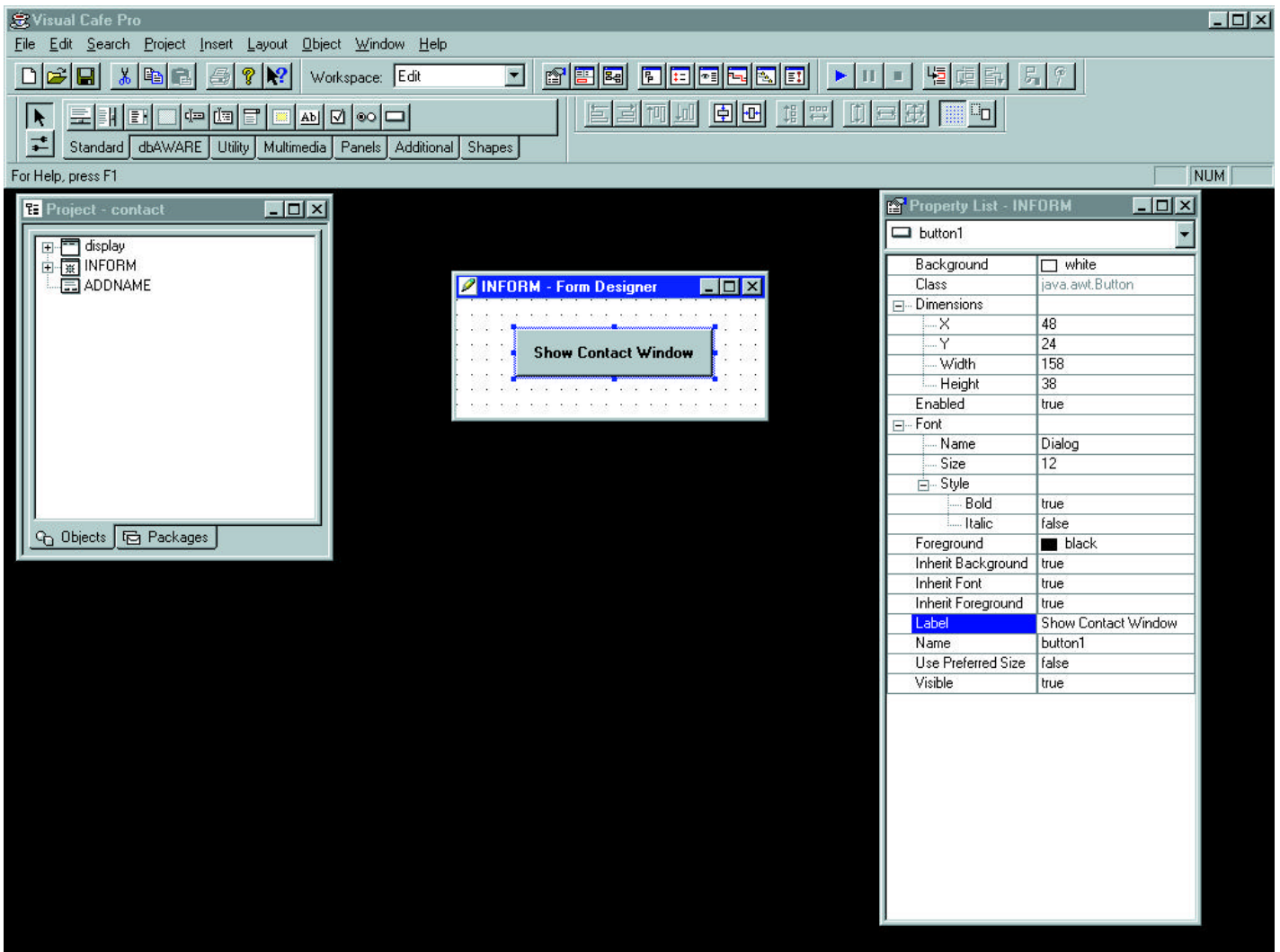


Figure 2

actual database is irrelevant. But, to show how easy it is to use, we will use an MS-Excel spreadsheet for the user records. In reality, this would be very inefficient for a large number of users. In future extensions, we will upgrade to an Oracle or Ingres database. The very fact that we can design the application like this shows the power Java holds: no restriction on platform and no restriction on the final database. By using dbAnywhere, or even the JDBC (Java Database Connectivity) API, upgrading the database does not require any code rewrites or modifications.

Storing this information in an Excel spreadsheet can be achieved by having each row represent a new contact name, with the first row describing the field names. Let's leave the database design for a moment and concentrate on what the user front end is going to look like.

Anyone who has designed any type of GUI (graphical user interface) using Java and the standard AWT package will know only too well the pitfalls that layout managers can be. "The devil's own creation!" some have been reported to say. Fortunately, Visual

Café Pro has taken over the responsibility of that particular nightmare and has made creating front ends a much easier task. For the remainder of this article, we will look at how Visual Café Pro can really help us design a front end applet for our Informer without writing a single line of code.

Informer Interface

As the name suggests, Visual Café Pro really is a visual experience. When you open up a new project, you are instantly presented with a Form designer window which is begging to have things dropped into it. Next to that window we have the Properties window that allows various parameters to be modified. For this application, we need to design an applet that will display a user's details and allow us to select from a list of other contacts.

With implementation being in an applet, we have two options to choose from; it can run within an HTML page, fixed inside a browser, or it can be in a floating window of its own. For this application, the floating window will be employed so the user does not have to keep their browser window

open to use the applet once it has loaded. But, in order to keep presentation of the applet simple for the user, the applet that will be placed in the HTML page will consist only of a button that will open up the main Informer window.

Setting this applet up is very easy. Insert a new applet component by selecting it from the menu option 'Insert'. This will bring up a small dialog box consisting of at least two icons. Choose the Applet icon. Clicking on the new applet component will bring the Form designer window for that applet. Here, we simply drag a new button to the center of the window and change the name of it by selecting the 'Label' properties field in the Properties window.

Next, we need to create the shell window for which all of the user information will be displayed. The class on which the popup window is based is the java.awt.Frame from the AWT package. Creating a new frame class is as simple as creating the first applet window. Select the Insert menu option and this time select the 'Component..' item. This will bring up a small dialog window that contains a list of

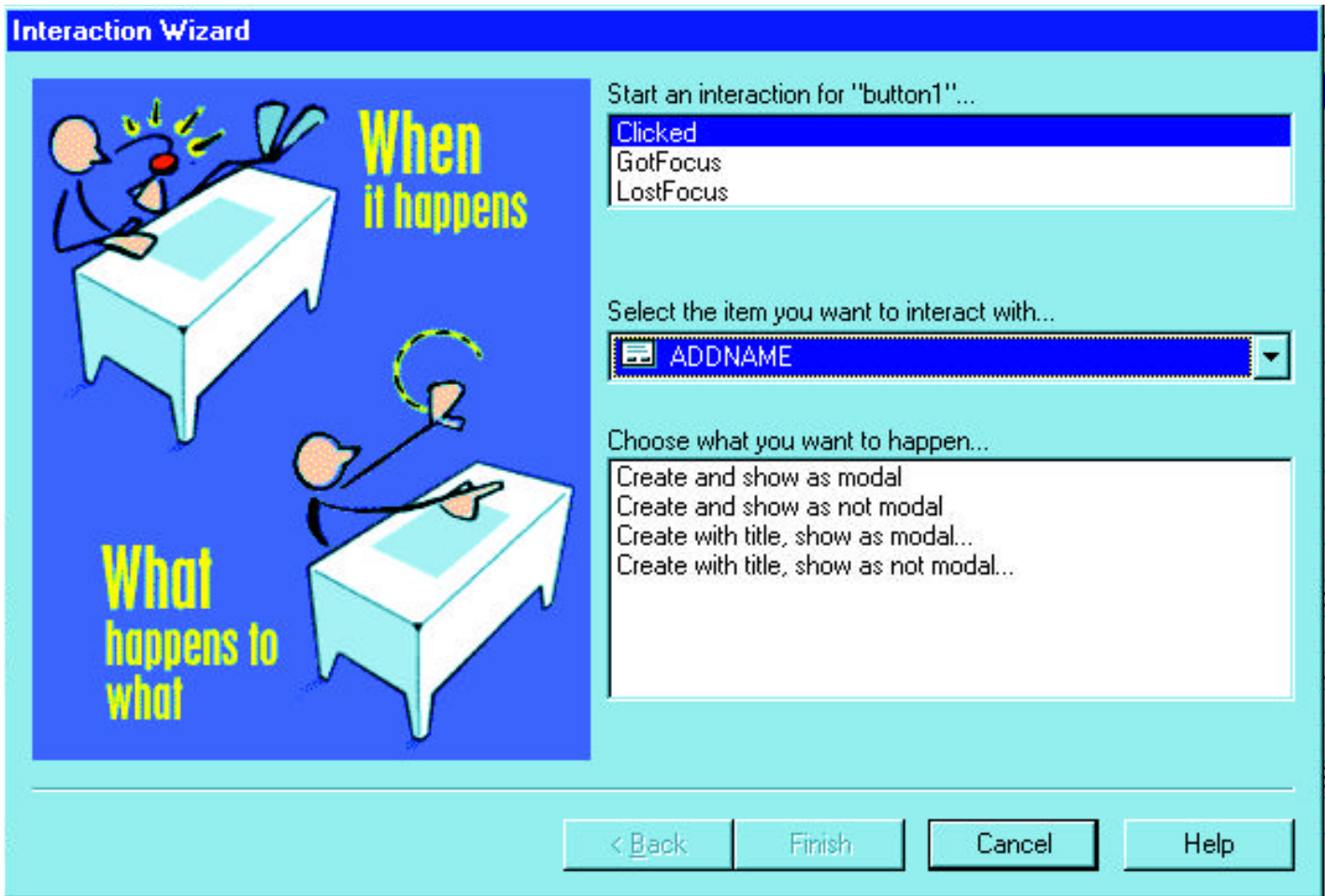


Figure 3

all of the different types of GUI objects that can be inserted. Move down the list until you get to the Forms section and select the Frame option. This will create a new frame object and display the Form designer window for it. Using the Form designer, objects can be dropped into the window and placed to your heart's content.

Interaction

When the user clicks the button in the applet, we want the main frame window to be displayed. We can achieve this in one of two ways: First, look through the documentation and work out how we can create and display a frame window when a user presses the button. Or, you can use something a lot nicer: the Interaction Wizard. This allows simple interactions to be set up just by dragging a line between two component objects. Let's create an interaction between the button and the blank frame window so that when clicked, our new window will be created and displayed.

First, open up the form with the button on it which will be used to open up the window. Next, click on the icon with the two electrical plugs; this will change the mouse cursor. When you move the cursor over various objects, they will highlight. You

need to select the origin object from which the activity or event will take place. In this case, click on the button in the main applet window. This will anchor a line from this object.

You have to select which object the interaction will work with. We want the frame to be displayed so we drag the line from our cursor to the frame object in the Project window. Having successfully chosen both objects, the dialog window shown in Figure 3 is displayed. From here, the nature of the interaction is described. As you can see, many different events can happen for a button, but in this instance we only wish to assign an interaction to the on click event. Having selected the event from which the event will be caught, we need to determine what will happen. This is selected from the bottom half of the window.

In this instance we want to display the window, which we can choose from the list. Now, close the dialog box. The interaction has been created. We can test this by hitting the play button on the toolbar or by selecting the Run command from the menu. This will run the applet with the button and, by clicking the button, if all is well, the blank frame window will be displayed. All this occurs without your looking at a single line of code.

Looking at the code Visual Café Pro produced won't do any harm at this point. This can be done by selecting the object, pressing the right hand mouse button and choosing the edit source menu option. This will open up the editor window, from which you can see how the Java code is working to produce the effect. You will notice how the code is also documented. How much better can it get? Not only does it write the code, but it documents the file as well!

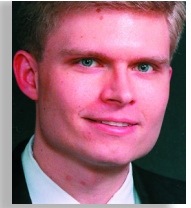
Conclusion

This column demonstrated the ease with which Visual Café Pro can be put to use doing the real coding and how, along with various wizards, it can make an excellent training companion. My next column will look at setting up the dbAnywhere database connectivity along with displaying the fields from the name. Future columns will add even more interactivity and functionality to the Informer project. ☘

About the Author

Alan Williamson is currently co-authoring a book with his colleague, Ceri Moran, on Java and databases, while running his Java-based company in the UK. He can be reached at www.n-ary.com or

 www.n-ary.com or alan@n-ary.com



Using JacORB

Pass by value as a handy tool for implementing distributed applications.

by Jeff Nelson

Distributed object solutions to date have relied on pass by reference for method arguments. However, pass by value results in a much more scalable solution in some applications which manipulate numerous, fine-grained objects. This article examines how to pass by value with CORBA. Then, this implementation is used to extend the RMI on CORBA design presented in the previous issue.

Introduction

Distributed object technologies, such as CORBA, allow developers to write applications in which objects in one program invoke methods on objects in another program. Thus, this greatly simplifies the task of writing client/server software.

However, argument passing for complex types, such as objects, doesn't work in quite the same way as for the primitive types and structures. At present, objects are always passed by reference in CORBA. For example, if you were implementing a BankTeller system and the client instantiated an instance of a Customer object, the client could pass a reference to its instance of the Customer object. The BankTeller could then manipulate the Customer object, invoking its methods and accessing its member variables using this reference. The instance would always be confined to the original CPU where it was instantiated, though. Other systems could manipulate the object through the reference, but not own their own copies of the object.

Passing objects by reference permits many different clients to manipulate large, complex objects in an efficient way. Moving instances of objects representing large data repositories or nested business objects

would quickly grind a network to a halt. By passing only a reference to an object, which in CORBA is represented by a string of text approximately 200 bytes long, large objects can be used in a much more scalable manner.

On the other hand, passing objects by value has benefits as well. For example, some objects, called transient objects, are only accessed by one client. In such a situation, keeping the object on the server across a WAN may not make sense in comparison to sending the instance of the object directly to the client. Also, some objects are very small, perhaps only a few bytes in size. If such objects are actually smaller than the roughly 200 byte size of the network representation of a CORBA object reference, then a performance gain may be realized by actually copying the value of the object rather than using an object reference. Finally, local instances of objects may be manipulated more efficiently than objects that are remote just because of network overhead.

These motivating forces lead to the Objects-by-value service in CORBA, which is expected to become part of the CORBA standard by the time this article is published. Objects-by-value allows method invocations to occur in which the objects are passed by value rather than by reference. Visigenic's Caffeine is an example of a CORBA distributed object development tool with support for the new Objects-by-value service.

Roll Your Own

While many will opt to purchase Caffeine, writing pass by value for yourself is not difficult. In the last column I examined how JacORB, a free CORBA development tool, could be used to implement a simple

Since the dawn of the CORBA specification, distributed objects have offered a powerful way to connect networked applications and services. Distribution across heterogeneity at many levels is hard, but that's where CORBA excels. With a powerful way to define interface (CORBA IDL) and a straightforward messaging model (requests), CORBA defines the standard way to bridge diversity. Unfortunately, the static nature of CORBA standard objects has, however, gotten in the way; CORBA never addressed "calling by value" (that is, passing objects over the wire as parameters of requests). In the last issue, Jeff Nelson walked us through a way to implement the JAVA RMI using CORBA; this time, he shows us how to add the Java "call by value feature" in a CORBA ORB for JAVA RMI applications. In light of the recent Sun Microsystems announcement that RMI will now use CORBA to implement Java distributed objects, this article is quite timely.

Richard Soley
Editor, CORBACORNER
Vice President and Technical Director of the
Object Management Group, Inc.

RMI clone with CORBA compatibility. Now, let's look at how JacORB could be used to implement passing objects by value. The benefit of using JacORB rather than a commercial product is that all the source code is available for you, the reader, to try out for yourself. Just download it from my Web site mentioned at the end of this article.

Let's consider an example of a problem in which pass by value may be useful. Suppose you were asked to implement a distributed point of sale system for a grocery store. One obvious object that your solution would almost certainly include is an Item object representing an individual grocery item available for purchase. The grocery item might be as simple as an

object with a few members, such as the name and cost of the item. A sample of an implementation of the Item object is shown in Listing 1.

The grocery store system might also include a Bag object representing the grocery bag of each store customer. The Bag object would have a method such as getItem() for obtaining the next grocery item, represented by an Item object, in the customer's grocery bag. For the purposes of this article, let's say the Bag object is limited to just that much capability and is represented by the following interface.

```
public interface Bag
{
    public Item getItem();
}
```

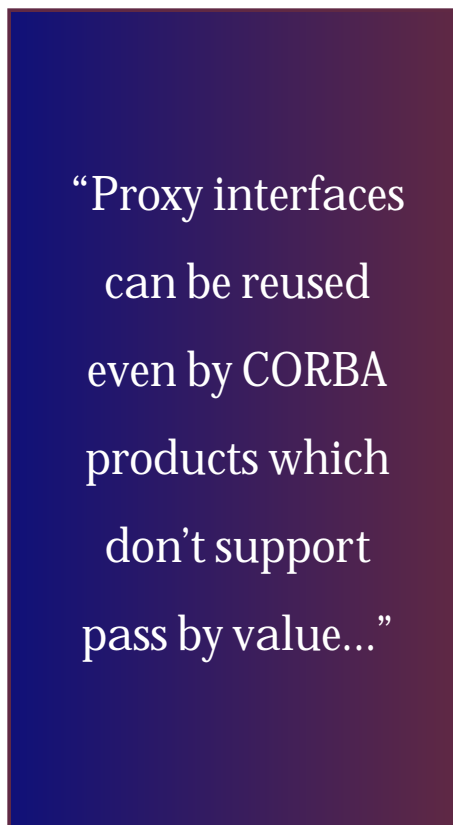
Consider this problem for a moment in the context of argument passing. The Item objects are small and numerous. Each client is likely to have many of them, potentially hundreds. If pass by reference were used, all of the instances of these hundreds of Item objects for every client would have to live on the server. Clearly, this could introduce problems with scalability since many clients communicating with one server might easily instantiate tens of thousands of small Item objects. This problem is clearly a candidate for pass by value in order to optimize performance and increase scalability. Let's tackle it by building pass by value using JacORB.

JacORB and Caffeine are among a handful of tools which supports the compiling of Java interfaces into distributed application stubs and skeletons. Typically, a CORBA developer would have to write Interface Definition Language (IDL), but this requirement is eliminated when using tools such as JacORB. However, unlike Caffeine, JacORB does have one limitation. It does not allow instances of Java objects to be passed in methods. For example, in the Bag interface, getItem() returns an Item object which is an instance of a Java class. Without the Objects-by-value service, which is implemented in Caffeine but not in JacORB, an instance of the Item object cannot be returned.

As a first step, then, let's wrap the instance of the Item object within a remote proxy which is a legal CORBA interface. Let's name this proxy ItemProxy and give it a few accessors for its member variables. The resulting ItemProxy is shown in Listing 2. In practice, you might envision a compiler generating all of this code based on the results of parsing the Item interface.

The Bag object must have a proxy of its own since returning an instance of Item makes the original Bag interface illegal for CORBA without Objects By Value. The BagProxy interface is identical to the original Bag interface but returns an ItemProxy object reference from getItem() rather than an instance of Item. By the way, the Proxy design pattern was documented very well by Hans Rohnert in "The Proxy Design Pattern Revisited". We also need to implement these interfaces with classes that wrap the corresponding instances of Item and Bag. For example, BagProxyImpl would contain members such as those shown in Listing 3.

Implementing ItemProxy requires a little bit more than simply wrapping an instance of Item. In this case, the Item object must



have the added capability of moving from place to place, so some capability must be provided for externalizing the Item object, copying it across the network and internalizing it on the other side. The CORBA Externalization service could be used to implement this. In fact, the architecture of CORBA Externalization is very similar to Java Object Serialization. However, for simplicity, I prefer to just copy the Item object into a byte array and pass the byte array. The implementation of this is available with the downloadable source code. Due to space restrictions, it could not be reproduced here.

At this point, we have passed all the real road blocks for writing our own pass by

value. The only remaining question is how to list the Bag instance with a name server for the world to see. Since the Bag object is not a legal CORBA interface, the Bag is, of course, simply wrapped in the BagProxy interface by instantiating a BagProxyImpl. On the client side, the BagProxy is replaced by a stub implementing Bag using dynamic stub loading. This technique is really very simple despite its grandiose name. All this means is that on the client side an object implementing the Bag interfaces is constructed using the BagProxy and returned from the Naming service. The full details are available in the downloadable source code.

That's it! We are ready to write a client and server that uses pass by value. Our sample server client is very simple – just two lines of code that instantiate an implementation of Bag and list it with the Naming service.

```
BagImpl obj = new BagImpl("BagServer");
Naming.rebind("BagServer", obj);
```

The sample client is equally simple – three lines of code.

```
Bag obj = (Bag) Naming.lookup("http://localhost/BagServer");
Item anItem = obj.getItem();
System.out.println(anItem.name + "costs" +
    anItem.cost + ".");
```

Notice that in this sample client, values of the instance of the Item object are accessed directly. In other words, this parameter was passed by value!

The results of running this sample are just the following:

Pound of Coffee costs 4.25.

Nothing spectacular, unless you purchase a lot of coffee, but it demonstrates that we have successfully passed an object by value. The design of the grocery store distributed applications can now take advantage of this capability to construct a more efficient, scalable system.

One of the nice features of this implementation of passing objects by value is that the Proxy interfaces can be reused even by CORBA products which don't support pass by value. This is a significant gain since it allows existing Visual Basic, Ada and C++ tools to easily work with the new system. For example, with only a little extra work, we could also write a Visual Basic client for the grocery store example.

Pass By Value In Other Orbs

Some other non-CORBA compliant

Object Request Brokers also support passing objects by value. For example, in Remote Method Invocation (RMI), objects are categorized as either local or remote at the time the object is implemented. This is done by implementing the Java interface `java.rmi.Remote` to indicate that an object is remote. All other objects are treated as local objects. When determining what argument passing policy to use, RMI depends on these categories. Remote objects are passed by reference, while local objects are passed by value.

This design has several implications. First, local objects, such as a Java Vector, cannot be passed by reference. Java developers, who are familiar with passing Vectors into methods, having the Vector modify the method and then looking at the results when the method returns, may be concerned about this.

Second, many applications have objects which should be treated as local or remote depending on the context in which they are used. Since the determination of whether an object is local or remote with RMI is decided at the time the object is implemented rather than at the time the object is used, these applications will actually require the maintenance of two independent object hierarchies. This could mean the need to implement two complete copies of all the objects in the entire object model, doubling the resource requirements of a project.

Finally, RMI's pass by value is based on Java Object Serialization, which is not well specified. This feature more than anything else stands in the way of interoperating the existing Java RMI with other programming languages, such as C++ and Visual Basic.

In last month's column, I examined how to build many of the features of RMI on top of CORBA, resulting in allowing RMI to interoperate with other CORBA tools and many different programming languages. In fact, the column showed a running example of an RMI application working with a Visual Basic client.

Despite these concerns about pass by value in RMI, this design of RMI on CORBA can now be extended to include pass by value. However, one firm requirement is that this implementation should not modify the way in which RMI developers write their applications, even how they use pass by value.

Let's go back to the grocery store example and use that as a test candidate to implement RMI's pass by value with CORBA. As a first step, let's write a pure RMI version of this example. The source code required to do this is virtually unchanged. Of course, the "generated" files such as all the Proxy and ProxyImpl

files are not required by RMI. Instead, RMI will provide its own generated files called "BagImpl_Stub.java" and "BagImpl_Skel.java". The Naming service implemented above is actually API compatible with RMI's Naming service. All you have to do is add the appropriate "import" statements at the top of the source code to include the Java RMI packages. Then, compile the example using the "rmic" compiler.

When you run the RMI example, the client should print out:

Pound of Coffee costs 4.25.

How do we implement this pass by value example on CORBA? Well, first just

“Passing objects by reference permits many different clients to manipulate large, complex objects in an efficient way.”

start with the same source code presented last month which implements RMI constructs such as Remote, RemoteException and Naming on top of CORBA. The only required addition is support for the new Proxy interfaces presented above.

Incredibly, all of these changes are restricted to the introduction of a new set of stubs and some slight changes to the Naming object. The Naming object is responsible for locating an object that implements the Bag interface. The dynam-

ic stub loading capability of the Naming object will handle the construction of a Bag stub based on the BagProxy which was passed to it. In order to handle pass by value, the Item stub includes a `fromByteArray()` method which internalizes an instance of Item. That's it!

Once these changes are made, you should be able to run the actual RMI example (shown just a moment ago) right on top of CORBA, even though it includes pass by value. One rather nice benefit of this new design of pass by value for RMI is that it is completely interoperable with our implementation of pass by value in CORBA as well as with older CORBA products which don't support pass by value. For example, if you took the time to write a Visual Basic client for the BagProxy interface, at this point you could go ahead and run this client with your RMI server!

Summary

Pass by value is a handy tool for implementing distributed applications. The CORBA Objects-by-value service is an example of pass by value in CORBA tools. I have demonstrated a rather simple example of how to implement pass by value using byte arrays and proxy interfaces. One of the benefits of proxy interfaces is that they allow excellent compatibility with CORBA tools which do not support the Objects-by-value service. Finally, CORBA Objects-by-value can be used to implement pass by value for other ORBs such as RMI, resulting in a high degree of interoperability between the two technologies.

Where To Go From Here

JacORB is available on the Internet at <http://www.inf.fu-berlin.de/~brose/jacorb/>. RMI and Java can be found at <http://www.javasoft.com>. CORBA standards, such as Objects-by-value, can be found at <http://www.omg.org>. Visigenic, the maker of VisiBroker for Java and Caffeine, can be found at <http://www.visigenic.com> ☛

About the Author

Jeff Nelson is a consulting architect with Dialogos Incorporated, a premier provider of CORBA consulting and training. He has eight years of experience in distributed computing and object technology. Jeff holds a Masters Degree in Applied Mathematics and industry certifications from Sun, Microsoft and IBM. He can be found on the web at <http://www.mind-spring.com/~corba/> and reached through e-mail at corba@acm.org



corba@acm.org

Listing 1: Representing a grocery item.

```
public class Item
    implements java.io.Serializable
{
    public String name = null;
    public double cost = 0.;

    public Item(String n, double c)
    {
        name = n;
        cost = c;
    }
}
```

```
private Bag delegate;
public BagProxyImpl( Bag delegate )
{
    this.delegate = delegate;
}
public ItemProxy getItem()
{
    return new ItemProxyImpl( delegate.getItem() );
}
```

Listing 2: A Remote Proxy for an Item instance.

```
public interface ItemProxy
    implements jacorb.CORBject
{
    public String getName();
    public void setName(String name);

    public double getCost();
    public void setCost(double cost);
}
```

Listing 3: Implementing the BagProxy.

```
public class BagProxyImpl
    implements BagProxy
{
```

JavaLearning Center 1/2 Ad

Java Performance Fastest on Sun Solaris

Organizations are implementing Internet strategies to speed up business cycles and gain a leg up on their competition. They are relying on the Internet "dial tone" – or Webtone – to enable global Web-based computing and provide them with the capabilities to run their day-to-day business faster, smarter and cheaper.

Committed to delivering the fastest Java performance in the industry, Sun has integrated Java performance technology into the recently announced Web-enhanced release of Solaris (v. 2.6), enabling companies to run Java applications at significantly higher speeds than they can on Windows NT.

One analyst said Sun's announcement is part of the company's plan to deliver the best server-side Java performance and make Solaris a major platform for running Java applications, since Microsoft is less focused on Java as a platform.

"Microsoft came out and was trumpeting for many months how fast their Java virtual machine operated, and one of the things Sun has been touting is that Java is for the enterprise," said Evan Quinn, Research Director for Java and Internet software for International Data Corp. of Framingham, Mass.

Sun Solaris is extremely reliable, highly scalable and can be accessed and used by devices connected to the network -- from smart cards to supercomputers -- including NCs, NetPCs, Windows computers and NT servers. That's why many companies rely on it to deliver the Webtone and make Web-based computing a reality.

And because it sends small, secure programs across the network, Java is perfect for companies moving to Web-based computing. With Java, software programs can reside on the network, not inside each PC, and Java can run on any computer -- Windows PCs to Macs to mainframes -- making the network and the server the heart of a company's enterprise. Quite simply, Java makes the network the computer.

Real-World Deployment of Java on Solaris

Forward-looking organizations with early access to the Web-enhanced release of Solaris are enjoying all the benefits of reliable Web-based computing while developing break-away business strategies.

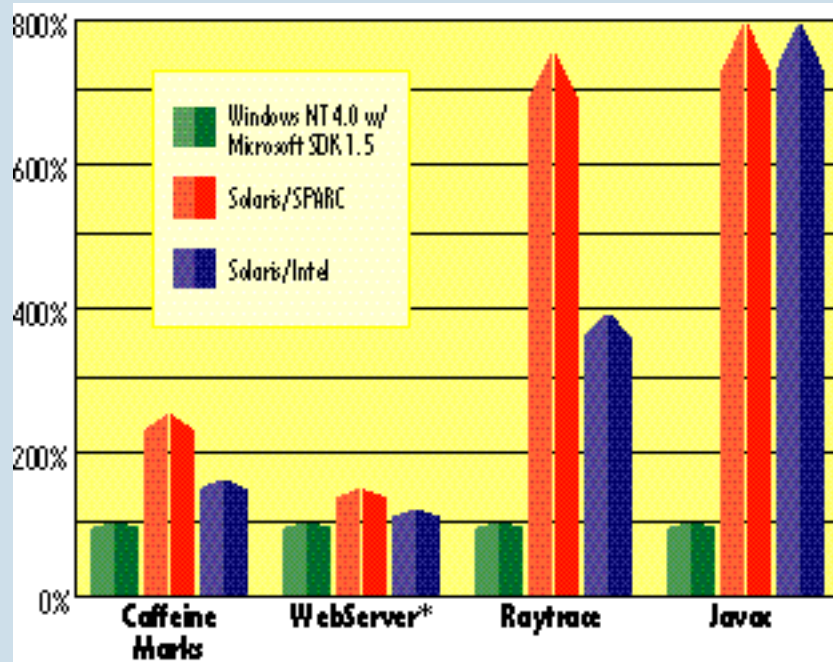
For instance, Vitria Technology, Inc. of Mountain View, Calif. and Volano LLC of San Francisco deploy Java applications on Solaris which enable them to 'Write Once, Run Anywhere,' reducing application development cycles while improving overall network computing performance.

"As a provider of enterprise caliber publish-subscribe solutions, Vitria delivers business-critical Java applications that are ultra-scalable to meet the requirements of our customers," said Carl Tsuka-

hara, Vice President of Marketing, Vitria Technology, Inc. "The Java integration with Solaris provides a robust combination that gives our customers the confidence that their Java servers will be highly scalable and highly available."

According to Mark Elderkin, president of Volano LLC, "VolanoChat™, a fully multi-threaded Java chat server, is used to provide live Internet chat services for tens of thousands of people everyday and is currently deployed in 14 countries." He continued, "On a system with native threads, our Java chat server can accommodate peak loads with minimal impact on CPU usage. Java on Solaris provides our customers with the stability and remote system management capabilities absolutely required but not readily available on other platforms."

Larry Weber, Vice President and General Manager, SunSoft, Inc. says Sun maximizes Java performance on real-world applications by coupling our knowledge of Java with our experience in high performance tools and compilers, and delivering Java on the industry's most powerful operating



*Windows NT using JDK 1.1.2 w/ Symantec JIT

Table 1

environment.

"Java on Solaris can generate millions of simultaneous operations at any given moment, delivering performance and throughput only Solaris can deliver. Today Sun customers, including Vitria Technology and Volano LLC, are experiencing the increased performance gains of Java on Solaris," said Weber.

Every Second Counts

Recent benchmark tests conducted by Sun show that Sun Solaris operating environment on SPARC and Intel platforms delivers the fastest Java performance in real-world applications, surpassing Microsoft Windows NT performance by more than 700 percent. Real-world Java applications are tested in industry Java benchmarks, including Java WebServer, Pendragon Software's CaffeineMark v2.5, raytrace, javac and others, to understand, test and measure true Java performance.

Fully complying to industry standards on Java, Java applications deployed on Solaris out-performed a Windows NT system by more than 700 percent in some benchmarks.

Windows NT using JDK 1.1.2 with Symantec JIT

- CaffeineMark: Measures the speed of Java

execution and simple 2-D graphics.

- Java WebServer: Measures the number of requests for static Web pages that a server can serve per second; important for high traffic Web sites and volume network-based transactions like airline reservation systems.
- Raytrace: Measures how systems perform with compute-intensive applications for sophisticated image processing.
- * Javac: Measures a large Java source file, indicating performance on large applications that require fast access to data.

For more information about Java on Solaris, Web-enhanced Solaris to be released in August, the latest Java Development Kit/Just-In-Time compiler, white papers and other valuable resource, visit <http://www.sun.com/solaris/new/java>.

Test Configuration/Methodology

For Java WebServer:

1. UltraSPARC, 2 x 200MHz, (Dual Processors), 1028 MB RAM.
2. Identical configurations were setup to test Solaris Intel and Microsoft NT 4.0: Pentium Pro, 2 x 200MHz (Dual Processors), 2 GB RAM.

For CaffeineMark, Raytrace and Javac: Sun Ultra Enterprise 2 x 200 Mhz Ultra-

SPARC 1028 MB RAM (but limited to 128 MB by software) Seagate Barracuda ST32171WC (2.1GB)

ALR Evolution Duel 6
2 x 200 Mhz Pentium Pro
128 MB RAM
Seagate Barracuda ST34371W (2GB) ●

Solaris at Work

- All the shipping traffic that is conducted in the Hong Kong harbor is managed via Solaris.
- The world's largest trading floor at one of America's largest banks depends upon the reliability and performance of Solaris to deliver reliable and secure service to its global traders 24 hours a day.
- Time, Inc. and Home Shopping Network selected Solaris to run their Web sites because of Solaris' performance, ability to handle heavy on-line traffic, reliability and Internet security.

1/2 Ad

Implementing a Security Policy

by Qusay Mahmoud

"Java is the language of network computing," said Lawrence J. Ellison, Chairman and Chief Executive Officer of Oracle Corporation. Basically, there are two main reasons as to why Java is the language of choice for network computing:

1. The APIs provided by the java.net package are very easy to use. Thus, the task of building client/server systems in Java is

seamless.

2. It provides an extensible security model that developers can extend to implement their own custom security policies for their client/server systems.

Security issues are a major concern in every client/server system. This is mainly due to the fact that the client will be inter-

acting directly with the server. Thus, the client will have access to all the resources and the ability to read files from the machine on which the server is running. In order to protect the server's machine from malicious clients' requests, there is a need for devising a security policy that states what the client can and cannot do on the server's machine.

Security in Java

The Security Model in Java is composed of three layers:

1. The Java language itself
2. The Java compiler and runtime system
3. The SecurityManager class

This article concentrates on the third layer. For more information on the other two layers, please see references at the end of this article.



Building secure client/server systems

The Java Language

Java achieves its safety in several ways. First, Java is strict in its definition of where all primitive types are guaranteed to be of a specific size, independent of the machine architecture. Second, you cannot do pointer arithmetic or forge access to objects. Third, it provides array-bounds checking. Thus, an attempt to index an out-of-bound

element of an array will throw an exception. Finally, Java ensures that a cast of one object to another is a legal operation.

The Java Compiler

The second layer is provided by the Java Compiler and Runtime System. This layer provides the necessary features to ensure that the Java system is not subverted by invalid code. This is done by providing a simple, secure execution environment that consists of the following sub-layers:

- The Java Bytecode Interpreter: Class format verifier
- A mechanism for dynamically loading and checking libraries at runtime
- Automated garbage collector

The Security Manager

The above two layers mainly ensure that the Java system is not subverted by invalid code. However, they do not provide any mechanisms to protect against malicious code in a network program. For a moment, imagine that a client is aware of a sensitive file (e.g., /etc/passwd) that is available on the Web (HTTP) server's host system. The client may request that file and, if a security policy is not implemented to protect against this, the server will be able to satisfy the client's malicious request.

This is just one kind of malicious instruction that the first two layers do not protect the server's host system against. However, given that Java code will adhere to the restrictions imposed by the Java runtime system, a security policy can be devised and implemented at the application level. This will allow us to state what sort of requests a Java client program can or cannot perform. For example, it will allow us to state if a client is allowed to read the sensitive file "/etc/passwd".

Building a Security Policy

The Java Development Kit does not come with a security policy mechanism that is ready for use by your applications. Rather, you have to define one yourself and implement it. However, the SecurityManager class, which is part of the java.lang package, provides the necessary mechanism for creating a custom security manager that defines tasks that an application can and cannot do.

The SecurityManager class is an abstract class (non-instantiable) that allows application developers to implement a security policy. The SecurityManager class provides you with a mechanism to establish a specific security policy that is suitable for the level of trust for a particular program. It provides the ability to create objects to determine if an operation that a program may perform is permitted.

The SecurityManager class provides

many methods for performing acts to enforce a security policy, some of which are shown in Table 1.

As you can see in Table 1, most of the methods in the SecurityManager class start with the word check. When your application uses one of the Java APIs to perform some tasks (say reading files), the methods that start with the word check are actually called by the Java APIs before performing certain sensitive operations (such as reading, writing or deleting files). If an operation is not allowed, a SecurityException will be thrown.

When writing a client/server application, make sure you devise a coherent security policy that states what the client is allowed to do with the server. For example, make sure that the server's machine file system is not at risk when performing clients' requests. Take into account any request that a client may perform. For example, if you are building an HTTP server, you must make sure that clients cannot get the file "etc/passwd" if they request it!

Building a Security Manager

Once a coherent security policy has been devised, implementing it is not particularly complicated. The SecurityManager class already provides abstract methods that we must override with new code that implements our security policy.

Example

As a realistic example of implementing a security policy and a custom security manager, let me remind you of the article "Programming an HTTP Web Server with JavaHTTP" (JDJ, Vol. 2, Issue 5), by Joseph DiBella. Note that this is not to criticize that article, but rather to show a realistic example where devising and implementing a security policy is a must.

DiBella wrote a nice article about how to implement a simple Web server in Java. Having written a Web server myself and taking into account that Security is a state of mind, the first thing I asked myself after reading Joseph's article was: "Is this HTTP server secure?" The answer was no. If you are up for some pure experimentation, try to compile the code and run it. Then, via your favorite Web browser (I use Netscape), try to fetch some documents that the server should not be allowed to ship to clients. For example, on my workstation, I ran the above server and I wanted to see if I could get the password file (which is normally located at /etc/passwd), so I constructed the following URL and gave it to Netscape to fetch: `http://myHostName:8080/../../../../etc/passwd`. The HTTP server found the file and shipped it back to the client (Netscape) and I got it

Method Name	Act
checkConnect(String,int)	Throws a SecurityException if the calling thread is not allowed to open a socket connection to the specified host and port number
checkDelete(String)	Throws a SecurityException if the calling thread is not allowed to delete the file
checkRead(String)	Throws a SecurityException if the calling thread is not allowed to delete the file specified by the string Note that the file can also be specified by a FileDescriptor.
checkExec(String)	Throws a SecurityException if the calling thread if not allowed to create a subprocess
checkExit(int)	Throws a SecurityException if the calling thread is not allowed to cause the JVM to halt with the specified status code
inClass(String)	Check if the specified String is in this class.

Table 1

displayed on the screen. Note that the number of "." depends on how far I am from the root directory and the countdown starts from the directory you ran the Web server from.

However, this is not all. A client's request may look like this: `http://myHostName:8080/etc/passwd` (note the extra slash after the port number). This says that I want to start from the root directory and get the file "etc/passwd", which is an easier way (than the use of "..") to get to the password file. Surprisingly, the server was able to deliver the requested file to the client.

A Security Policy

In a standard Web server, you certainly do not want people to be able to fetch documents that they are not allowed to. In other words, you want the server to have access to only a certain area of the file system, not the whole thing. In our case, we would like to prevent people from using "." as well as "/" as the first character of the requested file in their URLs.

Building Our SecurityManager

In order to create a secure environment for our Web server, we would subclass (inherit) the SecurityManager and override some of its methods. The SecurityManager class has various methods, such as: `checkAccess()`, `checkRead()`, `checkWrite()`, `checkConnect()`...etc. In our case, we would want to disallow people from constructing URLs that would permit them to read files which they should not be able to read. That is, as demonstrated above, we want to prevent them from using "." in their URLs, as well as "/" as the first character of the requested document. Thus, our new SecurityManager can be constructed, as shown in Listing 1.

It is important to note that when defining a custom security manager, you must override some or all of the permission checking methods depending on the policies enforced by the security manager. By default, all of the methods will simply throw a SecurityException, meaning that the operation is not allowed, with the exception of the `checkTopLevelWindow()` method which returns a boolean value. It is important to note that when you include a method, such as `checkAccess()`, with an empty implementation body, as shown in Listing 1, then that means the operation is allowed.

If, on the other hand, you would like to allow an operation but in a more restricted form, then you must provide your own imple-

mentation. For example, if your security manager allows clients to read files, then

it must provide its own implementation for `checkRead()`, as shown in Listing 1. The implementation provided overrides the existing `checkRead(String)` and allows the client to read files from the server, provided that the path to the file does not contain "." and the requested file does not start with a slash.

Installing Our Security Manager

Once a custom Security Manager is built, it is time to install it in our Web server. Installing it is as easy as adding a line in the `main()` body of the program as follows:

```
System.setSecurityManager(new ServeSecurityManager());
```

The `ServeSecurityManager` class, also shown in Listing 1, throws a security exception once a user is trying to read a file they do not have access to. Thus, we need to catch that security exception. This new exception must be caught in the `fileOpened()` method in the HTTP server code.

Conclusion

Java is the language of choice for building future client/server applications. Security is a major concern in every client/server system. The SecurityManager class, part of the `java.lang` package, is an abstract class that allows application developers to easily implement their own custom security policies. ☛

References

1. Yellin, F., Low Level Security in Java:
<http://www.javasoft.com/sfaq/verifier.html>
2. SecurityManager class.
<http://www.javasoft.com/products/jdk1.1/docs/api>

About the Author

Ousay H. Mahmoud is a Software Designer at a major networking company in Ottawa, Canada. He holds a Bachelor of Science degree in Data Analysis and a Masters degree in Computer Science, both from The University of New Brunswick, Canada. You can reach him at dejavu@acm.org



dejavu@acm.org

Listing 1: New Security Manager

```
class ServeSecurityManager extends SecurityManager {
    // the following operations are allowed:
    public void checkConnect(String host, int port) { };
    public void checkCreateClassLoader() { };
    public void checkAccess(Thread g) { };
    public void checkListen(int port) { };
    public void checkLink(String lib) { };
    public void checkPropertyAccess(String key) { };
    public void checkAccept(String host, int port) { };
    public void checkAccess(ThreadGroup g) { };
    public void checkRead(FileDescriptor fd) { };
    public void checkWrite(String f) { };
    public void checkWrite(FileDescriptor fd) { };

    // here goes more methods that we want to override...

    public void checkRead(String filename){
        if ((filename.indexOf(".") != -1) ||
            (filename.startsWith("/"))) {
            throw new SecurityException("403 Forbidden. No enough
            rights to read: "+filename+");
        }
    }
}
```




SuperCede Database Edition, v.1.2

by SuperCede, Inc.

Everything a Java developer may need

by Jay D. Smith



SuperCede has a picture of a man that resembles the comic book character 'Flash'. This character is highly representative of the SuperCede development environment and SuperCede 'Flash Compiler' technology that allows the developer to modify the code of a running application. This Java development environment allows the user to generate standard Java classes for portability or native executables for Intel-based PCs. SuperCede comes in three flavors: the Java Edition, the Java/Active X edition and the Database Edition. This review concentrates on the Database Edition. The Database Edition is not just a well-written and well-packaged development environment for the Java developer, but also opens the door for Visual Basic and PowerBuilder developers to develop applications in Java that can not only generate enterprise-wide applications for the Intel platform but for any platform the Java Virtual Machine is developed for.

Product Contents

The Database Edition comes on CD with all of the documentation included on the CD. SuperCede, Inc. also included references and advertisements for other SuperCede/Java-compatible add-ons and tools.

Documentation

SuperCede comes with three softbound manuals: a User guide, a First Look manual and a New Features guide. In addition to

the manuals, SuperCede Inc. included two tutorials in the form of Adobe Acrobat files. And like most Windows programs, SuperCede came with a context-sensitive Windows Help file. They also have added the "Java Quick Reference Manual" in the form of a Windows Help file to shore up that Java syntax. As far as documentation is concerned, SuperCede has added enough documentation to help the new Java developer become productive and for the experienced Java developer to stay productive. The only additional documen-

SuperCede Database
 SuperCede, Inc.
 110 110th Ave. Northeast, Bellevue, WA 98004
 Phone: 800 365-8553
 Fax 425 637-5886
 Web: www.supercede.com
 Email: sales@supercede.com
 Requirements: 48666 MH processor or higher; 24 MG RAM; Win 95 or NT ver. 3.51 or higher; CD-ROM; min. 50 MG disk space.
 Street Price: \$499.

tation I would have added is more detailed explanations of the included examples. Although SuperCede does include a Read Me file in the subdirectory of each example, a more detailed step-by-step explanation could help.

Installation

The installation process ran, as expect-

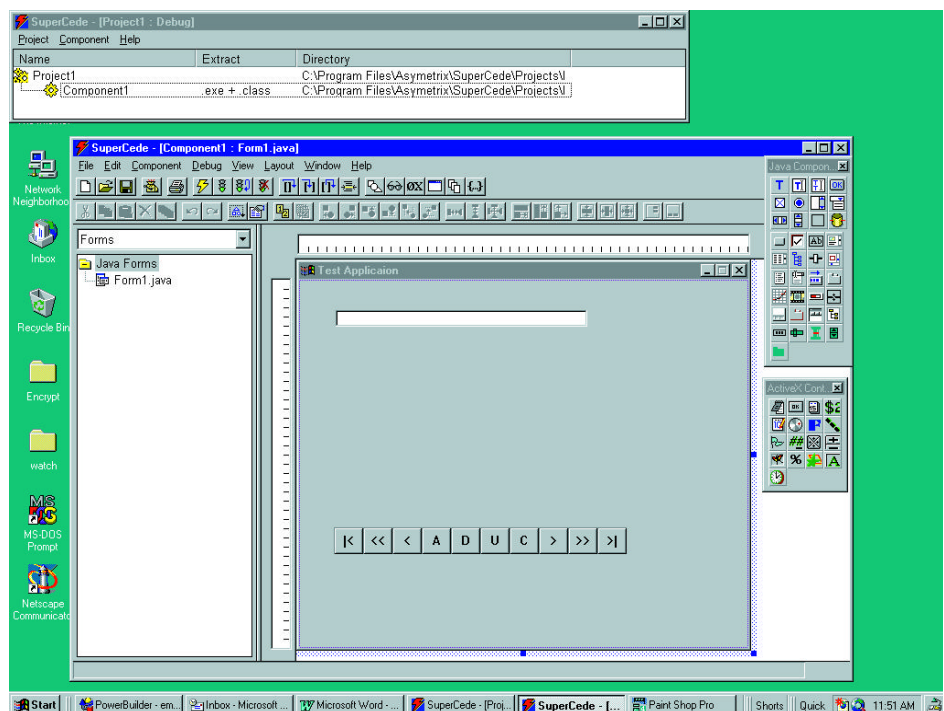


Figure 1: Project manager and main integrated development environment

ed, without a hitch. InstallShield was used to handle installation. The installation process was broken down into three parts: SuperCede Database Edition; MindQ Learning Java with SuperCede; and Starbase's Versions version control software. The process was intelligent enough to trigger environmental values on and off depending on the values. SuperCede Database was installed on both Windows 95 and Windows NT 4.0. Both install processes ran without a hitch, except that with the NT install the option to install the Visigenic VisiChannel for JDBC was available.

IDE

The integrated development environment is made up of the project manager and the main integrated development environment (see Figure 1). The project manager is used to show all the 'components' of a given project and their given file location. With the project manager, you can change properties on how a certain 'component' will be generated or extracted; i.e. class, DLL, exe, from Web page applets to full applications. The project manager also allows the developer to set up two environments to program for, one for debugging and the other for release. For example, a developer may want to have one environment to use one set of classes for debugging and another set of classes for the actual release version. SuperCede uses the idea of the 'components' metaphor loosely to describe the various parts that make up a project.

SuperCede's main integrated development environment, shown in Figure 1, is designed for 'at your fingertips' ease of use capability. As you can see, the menus and toolbars are well laid out in a methodological order similar to other Windows programs for user familiarity. The main environment comprises three major sections: the object browser, the form editor and the source editor. The object browser on the left side of Figure 1 allows for easy organization and visibility of the 'components' that make up your current project. Organization ranges from forms, source files, classes, external files, imported DLLs and content files to compilation order. Of particular interest to you OO people, the class browser will display all of the object's components in alphabetical order or hierarchy order along with the object's members (methods and properties). See Figure 2 for the object browser under Classes section.

The form editor is made up of the form and two palettes. It uses the drag and drop metaphor to allow for drag objects from the palettes to the form. The form includes two palettes: Java Components

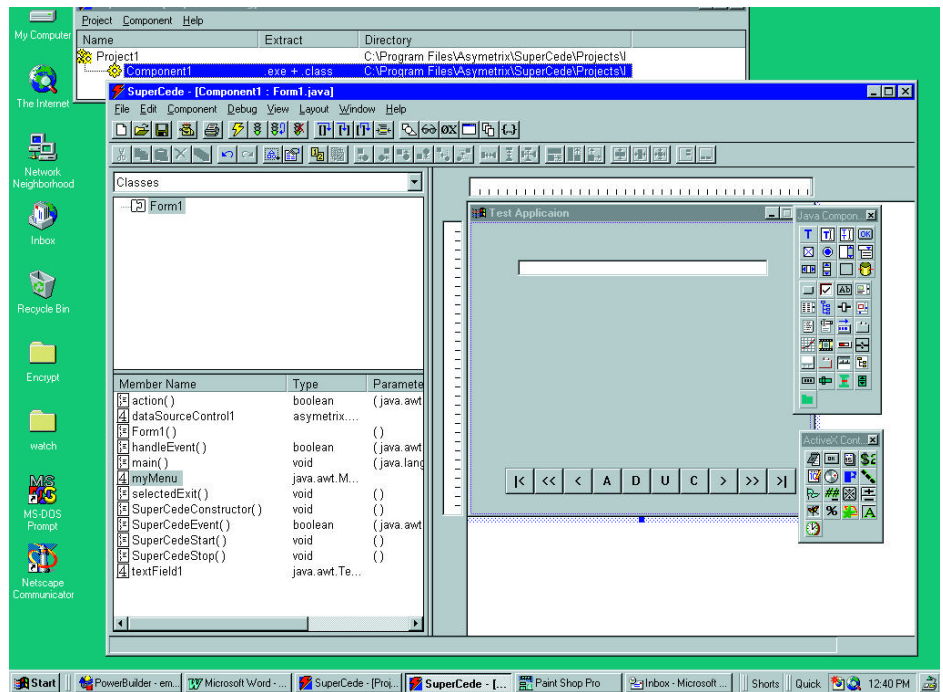


Figure 2: Object browser under Classes section

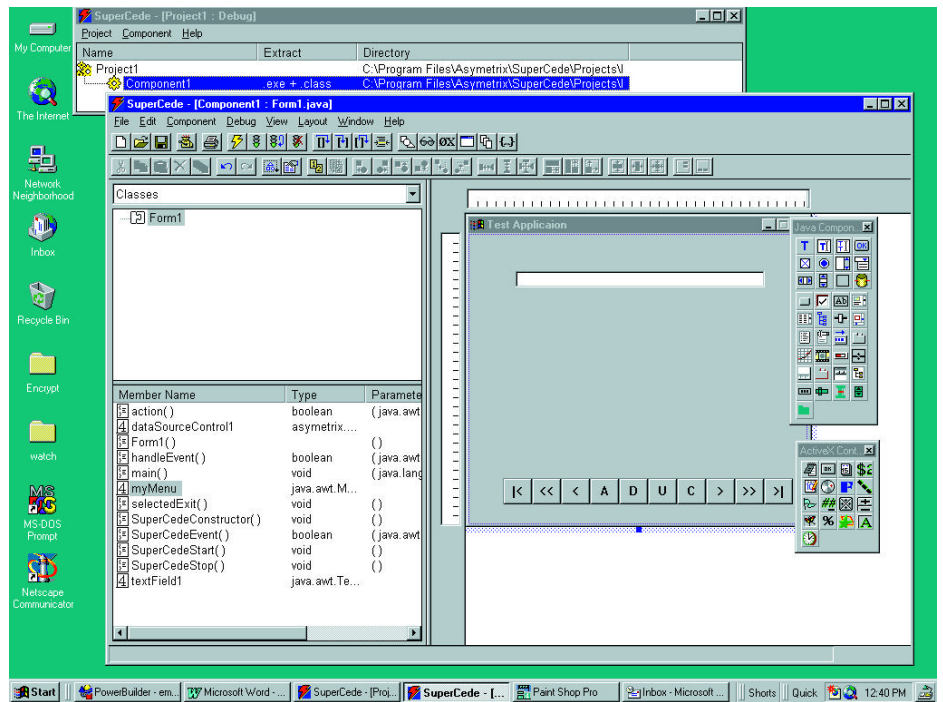


Figure 3: Accessible Properties

and Active X Components. Under the form, properties are accessible by double-clicking on the form or right-clicking and selecting the properties, options for styles, color and size (see Figure 3). The property window is of particular interest because it allows you to modify the properties of all of the objects on a form without having to close and reopen the property window. The file editor also includes easy-to-use navigation rulers and margins on the form with properties set up for uniformity.

The source editor is very easy to use. The actual code can be color coded to the developer's liking along with other options from a document level, display, colors (reserved words, comments, etc.), and find-and-replace. The editor is well integrated with the debugger. In fact, SuperCede, Inc. has added two well known features that Visual Basic users have come to love: toggling breakpoints while editing and adding code during a debugging session. Imagine all the times you were debugging and you have come across areas where it would be nice to

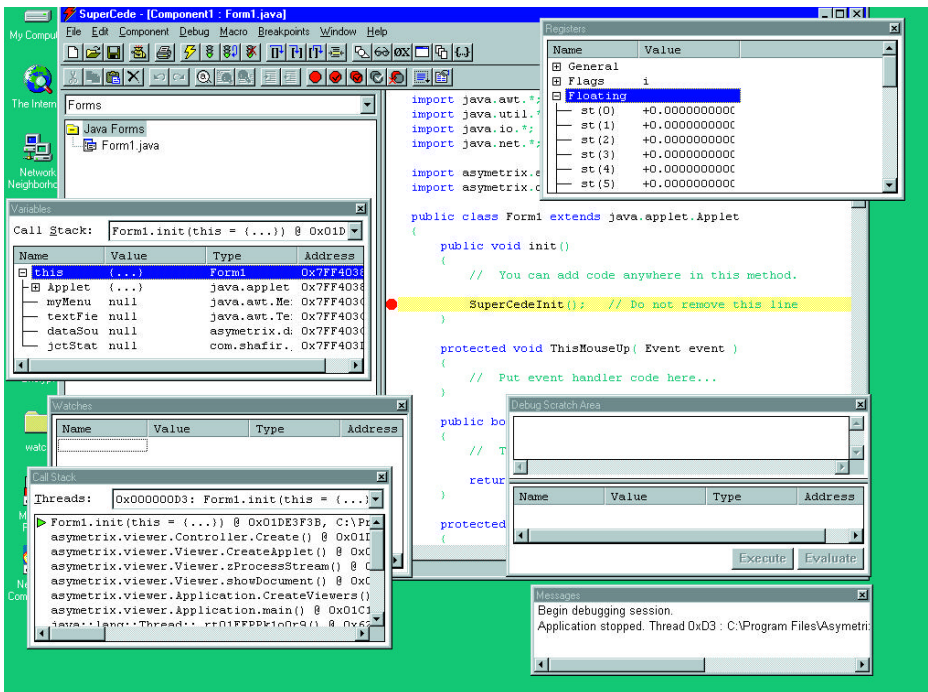


Figure 4

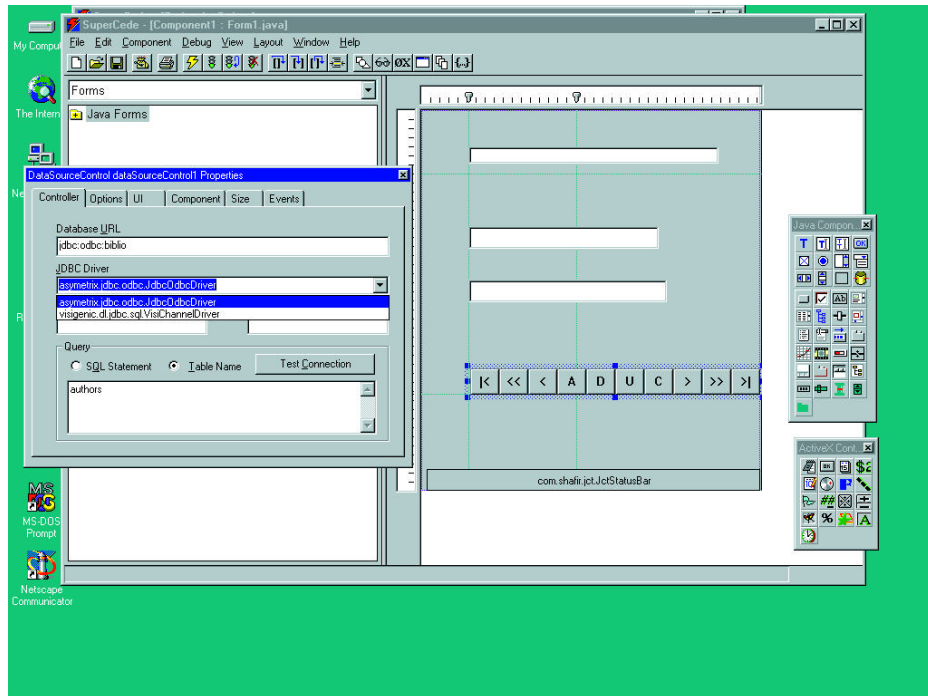


Figure 5

add some code and keep going from that point. We will talk more about the debugger's capabilities in the next section. Because the environment is so flexible, SuperCode allows the developer to use an external editor.

Debugger

I once read somewhere in a C programming book for Windows that developers spend 70 percent of their time debugging and 30 percent of their time doing the actual development. The interactive debugger

included in SuperCode's Database Edition helps decrease that debugging percentage time greatly by allowing the developer to edit while debugging. The interactive debugger also takes advantage of SuperCode's 'Pop' Technology and the fast compilation of source code. I rank this debugger up there with Visual C++ and Borland C++ as well as Visual Basic. The debugger includes various windows to view variables, watch specific variables, memory and registers. The debugger also includes a call stack to watch order of func-

tion calling. Unique to SuperCode, the 'Pop' Technology within the debugger allows the developer to restart a function from the beginning. With 'Pop' technology and editing while debugging alone, SuperCode can potentially double Java development.

Active X

The Java/Active X edition and the Database Edition of SuperCode include support for using Active X controls as components within a SuperCode project. Visual Basic developers will be right at home with using Active X controls here. In fact, to test this capability, I decided to use VB 5 to create a simple label control with masking capability. It worked like a charm. Just like VB, SuperCode supports drag and drop of the Active X control and custom property pages. Unlike VB, SuperCode does not support writing Active X controls, letting a Java program become an Active X document or the data-aware property possessed by some Active X controls. The New Features guide provides a step-by-step guide to using and registering Active X controls. In addition to this, an Active X tutorial is included.

Database Access

Although SuperCode's strong points are its IDE, debugger and Flash Compiler Technology, it also shines when it comes to database access. SuperCode Database Edition supports standard Java Database Connectivity. In fact, SuperCode includes the INTERSOLV JDBC-ODBC Bridge driver and the Visigenic VisiChannel JDBC Driver. In addition, SuperCode, Inc.'s includes a five-simultaneous-user version of Visigenic for JDBC Server for NT, which opens the door to three-tier programming and database access. With these capabilities, the developer can access not only desktop databases like Access and FoxPro but also enterprise databases like Oracle, SQL Server and Sybase. The New Features guide provides a step-by-step guide to using database capabilities. Aside from this, a database tutorial is included, along with other database access examples. SuperCode uses a special version of the jdbc.sql package that includes all the capability of the original jdbc.sql package and more. The jdbc.sql package includes support for the VisiChannel JDBC driver and Bridge driver. And if that was not enough, SuperCode Inc. added support for high-level database interfaces based on Microsoft's Active X Data Objects (ADO) specifications. The form editor's current Abstract Windowing Toolkit (AWT) is data-aware. Furthermore, a database source control (DSC) has been added to allow for easy set up of database connectivity.

Extras

SuperCede Inc. has added a lot of extras to make a great product even better. Some of them are Versions version control by Starbase and ObjectStore PSE for Java, which allows for persistent data storage. In addition, SuperCede includes a number of Java components such as tab controls, status bar, toolbar, masked edit, progress meter, slider, spinner, outline control, charting and many more Java components that can be used in SuperCede projects. And, SuperCede includes a number of Active X controls ranging from memo controls to time controls and many more.

To get the average programmer coming from other languages acclimated to Java, SuperCede, Inc. has included MindQ's multimedia Java training. Other makers of Java development environments should take notes from SuperCede's ability to make the transition from other languages to Java. In fact, for VB programmers, the jump to Java has been made even simpler with integration of TV Objects VB to Java conversion tool in to the SuperCede IDE. SuperCede includes the basic edition of the converter, with a discounted upgrade path to the professional edition.

The current version of SuperCede is version 1.2, which supports Java 1.0.2. However, by the time you read this review,

SuperCede will have a free upgrade path available which includes two major upgrades. One is support for Java 1.2 and the other is support for JavaBeans™. You will find information about the free upgrade at www.supercede.com.

Examples and Tutorials

SuperCede, Inc. has included two major tutorials; one is on using the controls that are included and the other is a database tutorial that shows how to use the database feature in SuperCede. The examples included range from basic text manipulation, Active X and OLE to simple game programming, appointment calendar, sorting, fantasy draft and many more. Although the two tutorials were very helpful in becoming familiar with SuperCede, I would have liked to see more detailed step-by-step explanations of the included examples.

Wish List

How can you improve on an already great product? Being the perfectionist, I found a couple of ways. One is to include a simple way to add properties, methods, and events to objects. Along the same lines, a more detailed graphical display of objects would be useful. Another improvement would be an easier way to modify menu objects, maybe a simple graphical

menu editor that generates code within a form. And finally, an easier method for creating the SQL code to the database control would be very useful, maybe in the form of a simple SQL Editor that can not only be use in the database control but also to preview and test SQL code. And while we are talking about a wish list, how about an integrated way to create add-ins? For example, developers may want to create wizards to use in their Java development endeavors.

Summary

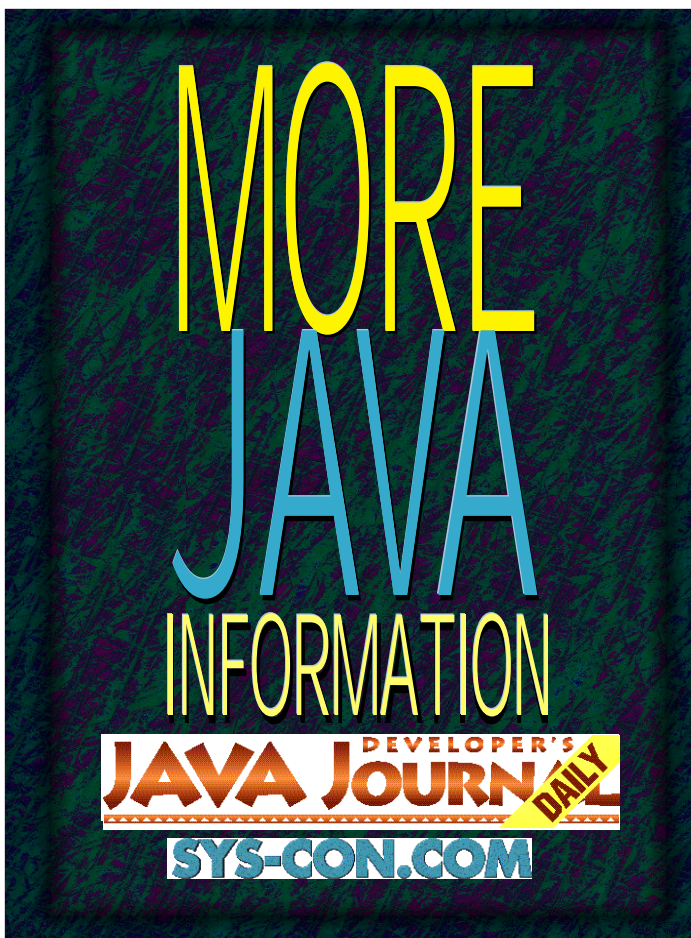
SuperCede Database Edition includes everything a Java developer may need. The Flash Compiler Technology, Active X support, great debugger, JavaBeans support and 'at your fingertips' ease of use raises the bar not only for just Java application tools but other Rapid Application Development environments as well. I highly recommend this tool. In fact, I recommend it so much that I am currently using SuperCede to do my own Java development. ☺

About the Author

Jay D. Smith is a consultant for Step Consulting, Inc. in Charlotte, NC. He can be reached at jayd@stepinc.com



jayd@stepinc.com



Critical Mass 1/4 Ad



Bazaar Analyzer Pro

by Aquas

A nicely executed idea, cleverly and purposefully designed, that should keep your marketing department happy

by Jon Steiner



In many companies, Web site statistics must be analyzed by many people throughout the organization. Your consulting vision may want to know

how often its page has been hit and by whom, while your software products division will want to know how often a particular page has been visited and how long each page was viewed (as well as who viewed it.). To generate this kind of information, it previously was necessary to ask the Webmaster to do so, which would normally entail either going to the server yourself (if you were the Webmaster) or requesting that they set aside some valuable time to generate the report. Many log analysis programs can be configured so that a specified report (such as most common domain name) can be e-mailed to a number of recipients. This is a nice feature, but if you're not the Webmaster and you want to request additional information, you'll have to annoy your poor, over-worked Webmaster again. How can you give people throughout your organization access to the information they need (especially marketing people with little technical background) and give them the flexibility they want (while at the same time keeping them out of your hair)?

There really was no way to do so until Aquas came out with Bazaar Analyzer Pro 1.0. Bazaar Analyzer Pro is a log-analysis software package for Windows NT and many variants of UNIX which enables people throughout your organization to generate the reports they need when they need them. You gain access to Web Analyzer Pro through a Java-based client GUI interface. The idea of a Java client and platform-spe-

cific server software is, in my opinion, an excellent design decision by Aquas which allows users anywhere on the Net the ability to create Web site statistical reports when needed, no matter which platform they choose to access this information from (or over an Intranet or an Internet.) If your marketing department uses Macs, and the Web server is a UNIX variant, you would normally need your Webmaster to generate a report for your marketing staff. Now you can do it yourself.

A variety of reports that are of interest to just about every area of a company are preconfigured: a Bandwidth Report, Most/Least Requested Pages, the

Bazaar Analyzer Pro
 Aquas
 599 N. Mathilda Ave., Suite 210
 Sunnyvale, CA 94086
 Phone: 408 737-7122 Fax: 408 737-1292
 Web: www.aquas.com
 Email: info@aquas.com
 Requirements: 32 MB available RAM recommended; 5MB free disk space
 Price: \$699 per license

type of organization which generates the most traffic (.com, .edu, etc.). One thing that particularly impressed me was the extensive support Bazaar Analyzer Pro has for country domains (.us, .nz, etc) If your company has an international presence (which could be any company on the Web) you will find this feature useful. Applied Visions, Inc. products have sold well in overseas markets, mostly thanks to our on-line marketing efforts, and we want to do everything possible to ensure

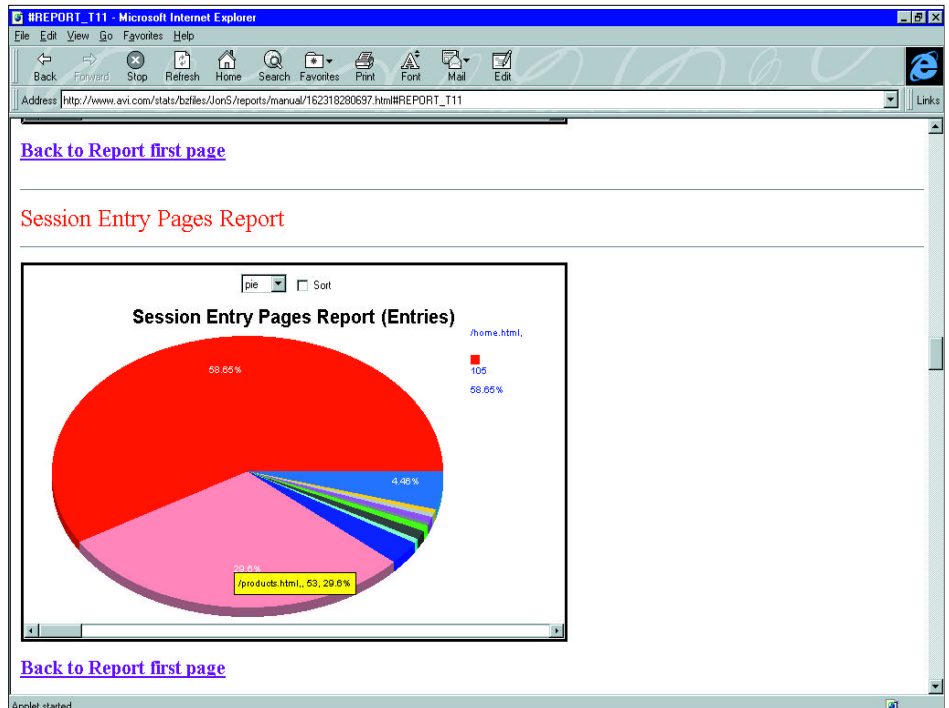


Figure 1: Report Generation Tool

that the content of our site will attract many more international customers. This feature lets us cater to our international clientele because we know who they are.

Bazaar Analyzer Pro 1.0 consists of five major parts:

1. An Administration Tool
2. A Report Configuration Tool (which reports you wish to run)
3. A Report Generation Tool
4. An Options Menu which consists of three tools:
 - a report generation scheduler
 - an Alert function which lets you spy on a particular page and if a particular user has hit that page
 - the Bazaar Minder, which lets you monitor another URL if its contents have been changed
5. A Watch Tool generates real-time statistics over your Intranet/Internet as to your site's use. This would be helpful to let you monitor how effective a media announcement is in driving traffic to your site, for instance

The Report Generation Tool

The Administration Tool is simple for anyone familiar with creating user accounts on a network. You need to set a password and let Bazaar Analyzer Pro know where to find the log files. That's as far as it goes.

The Profiler (in the Customize menu) is the tool which lets you configure the type of report you wish to run and specifically how you wish to see your data represented. (Do you care that 1 percent of your traffic is from *.mil Web sites? Or, would you rather have it represented in the "Other" section?) You can configure it from this menu.

If you expect several people to need access to a given report, you may archive it and they may retrieve it at their convenience.

The Report Generation Tool generates reports in HTML, pops another browser and displays the information in Java applets. Each applet may be dynamically rerendered as a pie, bar or line graph, or as a table. The applet dynamically explains information when the mouse is moved over an area for clarification.

Here are some of the reports that are generated:

The Options Menu

The Report Scheduler

The first utility in the Options menu, the report scheduler lets you specify the type of report to generate, the frequency with which to generate it and the people to e-mail it to, if desired. I thought this feature to be good, but what is really the point of e-mailing an HTML-only non-interactive ver-

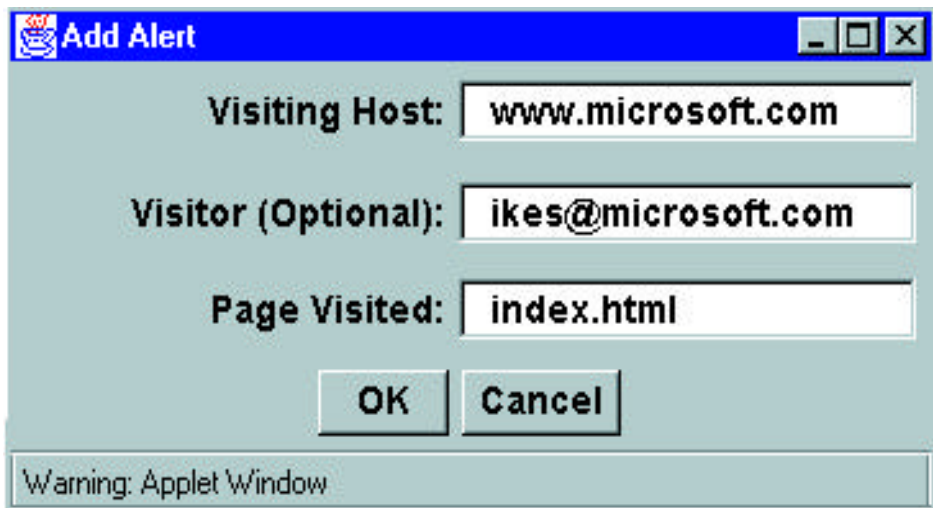


Figure 2: The NetDynamics Studio



Figure 3: The Minder

sion of a report to someone when they can view the fully interactive Java-based version? This is a standard feature in most log analysis tools, which is probably why they included it, but this should have been an opportunity to provide better reports than other Web site analysis tools.

The Alert Feature

If I'm trying to sell a client my latest product and I direct him to my Web site, how do I know if he ever came to visit? Your average executive normally would have to ask their Webmaster to search the Web server's log or the customer's URL in a generated report. Bazaar Analyzer Pro has an Alert feature which pops up in your window when a particular user visits a page. It can be configured for multiple users or without specifying a user. I thought this feature to be especially nice because it allows people who may not be familiar with Web servers and their tools to use the corporate server as a business tool.

The Minder

The Minder monitors a URL at a remote

location and e-mails you when something on that page has changed. This feature is far more efficient than an application like FrontPage, which pushes you to check all the links in your Web with a tool. A simple e-mail from Bazaar Analyzer Pro will enable you to maintain all external links on your Web site, as well as monitor certain URLs for product updates (including your competitors).

The Watch window allows for real time monitoring of the number of hits a page is getting. You may change the style of the graph, the page, the scale of the graph or the increments in which the information is generated.

Overall Analysis

What the Bazaar Analyzer Pro does, it does well. It will provide your organization with the data necessary to make informed decision about the organization and nature of your Web site. It also will enable your entire organization to use your Web site as a tool for business. I would say that this is a great example of the type of application which greatly ben-

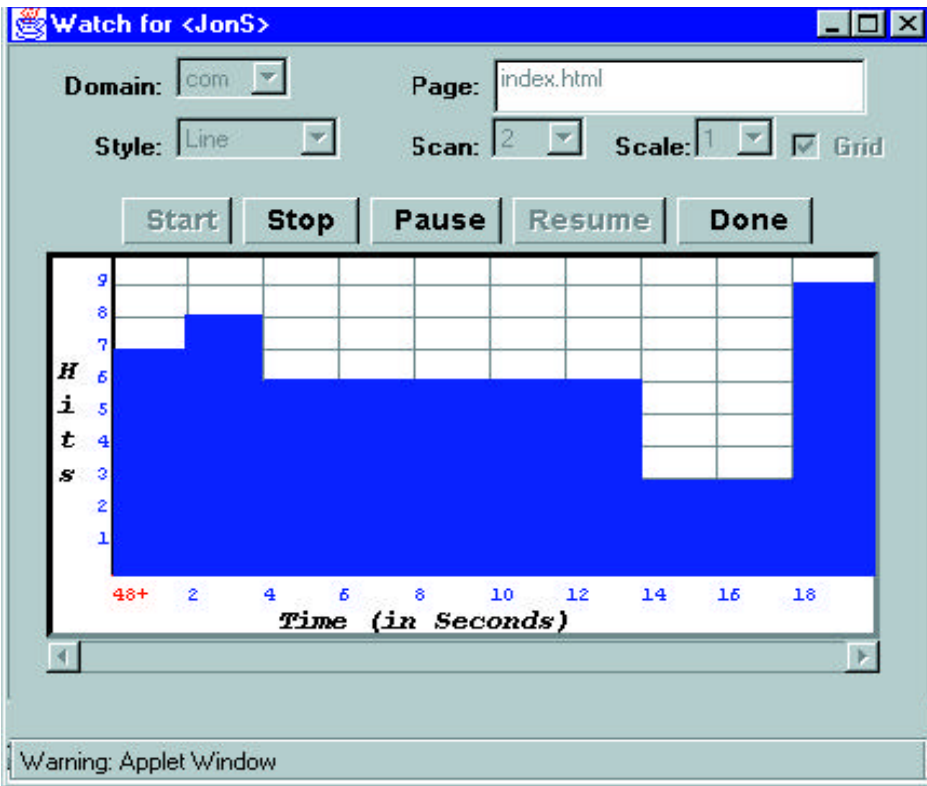


Figure 4: The Watch Window

efits from Java – any user could use all the features of Bazaar Analyzer Pro – whether the user was on a UNIX or Windows

machine. If this sort of flexibility is important to your organization, you may want to download the latest copy at

<http://www.aquas.com>.

One caveat: I had numerous problems installing Web Analyzer Pro 1.0. Aquas was very patient with me, offering generous assistance. By the time you read this, Web Analyzer Pro 2.0 should be out. It should contain, among other things, historical tracking of Alerts, the ability to track the path of a user through your Web site and an option to interface your Web server to a pager. Hopefully, in the next release Aquas will have streamlined their installation and configuration process.

Web Analyzer Pro is competitively priced at \$699 per license. If you're a Webmaster in an organization for which multiple people in different parts of that organization (or different organizations entirely) need access to information, and need it on demand, I urge you to try Web Analyzer Pro. It's a nice idea, nicely executed and cleverly and purposefully designed. And, it should keep your marketing department happy, which alone may be a good reason to try it. ☺

About the Author

Jon Steiner is the Product Manager and technical writer for ObjectGraphics for Java at Applied Visions, Inc. He can be reached at JonS@avi.com



JonS@avi.com

1/2 Ad

WebLogic Announces 100% Pure Java Server Components for the Enterprise

(San Francisco, CA) - WebLogic, Inc. has announced BeanT3™ and RmiT3™, 100% Pure Java™ components that enable Java™-based distributed computing across the enterprise. These products deliver the scalability and performance needed for mission-critical applications operating in Internet/Intranet and enterprise computing environments.

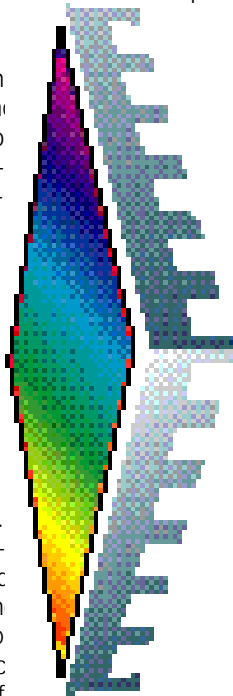
Bean T3 and RmiT3 are the industry's first commercial implementations based on, and completely compatible with, the corresponding JavaSoft standards. BeanT3 transforms off-the-shelf server JavaBeans™ into distributed client/network components, allowing transparent execution across multiple JVMs. RmiT3 is a high-performance, scalable implementation of JavaSoft's RMI specification.

Bean T3 and RmiT3 are in public beta now and will ship in September. Pricing is still to be determined. For more information, see WebLogic's Web site at www.weblogic.com or call 800 932 5644. ●

Sun and Microware Agree to Offer Comprehensive Package of Java APIs

(Des Moines, IA) - Microware Systems Corporation, developer of the OS-9 real-time operating system, has announced an expansion of Microware's license with Sun Microsystems, Inc.'s JavaSoft business unit which includes the Java platform for corporate applications such as personal and network computers; PersonalJava for more consumer-oriented devices such as digital TVs and mobile phones and EmbeddedJava for devices with limited resource requirements such as pagers, printers and industrial controllers. Microware will be the first embedded real-time operating system company to hit the market with this new Java package for OEMs.

Microware's latest licensing agreement with Sun will couple OS-9's unique attributes of flexibility, modularity and scalability with the powerful and intuitive Java programming language. With the broadest choice of options and the most extensive offering of I/O and interoperability across all networks, Java for OS-9 will provide an



optimal solution for embedded devices and broaden Java's uses in the consumer products market. Developers will be able to bring their products to market quicker because Java for OS-9 allows them to pick and choose exactly the components to include in their particular consumer product.

"Thanks to our close relationship with Sun, Microware is the first RTOS vendor to offer Java in a customizable package for use in virtually any product aimed at the consumer market," said Ken Kaplan, president of Microware. "By teaming with JavaSoft and combining the two industry-leading technologies of Java and OS-9, we have successfully moved Java off the desktop and into the high-growth, high-volume consumer products market."

For more information on Microware, please visit their Web site at www.microware.com. ●

Microsoft Development Tools Set New Milestones

(Orlando, FL) - Microsoft Corp. has announced that the Microsoft™ Visual Basic™ programming system and the Visual C++™ development system - Microsoft's industry-leading development tools for the Windows™ operating system - posted record revenues in March, while shipments of the newly released Visual Studio 97 development tool suite are exceeding expectations. In addition, the Microsoft Visual J++ development tool, Visual Basic and Visual C++ were identified as clear leaders in their categories by wide margins, according to an independent research study conducted in March by Market Dynamics Corp., a Portland, Oregon-based independent research firm that regularly tracks professional software developers.

Visual Basic and Visual C++ posted record revenues for the months of March and April, significantly higher than in any single previous month since the products were introduced. In addition to strong

sales, usage of Visual Basic and Visual C++ by professional developers is increasing rapidly. Visual Basic is both the most widely used and the fastest-growing development language in the world, with more than 3 million active developers worldwide. Leading its category, Visual C++ is used by 73 percent of professional developers using C++.

The study also reveals strong momentum for the Microsoft Virtual Machine for Java, evidenced by the fact that 96 percent of developers who use Java are developing and deploying their Java applications on Windows.

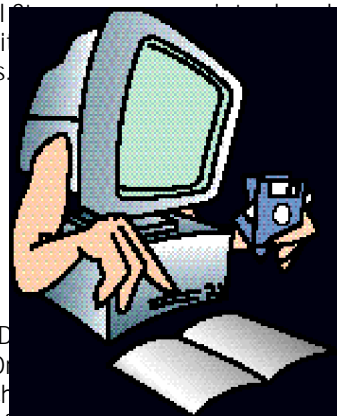
Both the Professional and Enterprise Editions of Visual Studio have quickly gained momentum in the retail channel since they were shipped in March. Visual Studio 97, Professional Edition, includes Visual Basic 5.0, Visual C++ 5.0, Visual J++ 1.1, the Visual InterDev+ application development system 1.0 and the Visual FoxPro+ database 5.0. The Enterprise Edition also includes Microsoft Transaction Server, the Visual SourceSafe+ version control system and Microsoft SQL Server+ 6.5. In addition to record-setting individual shipments for Visual Basic and Visual C++, retail outlets are already running out of Visual Studio 97.

If you are interested in additional information about Microsoft, visit their Web page at www.microsoft.com. ●

MindQ Makes It Easier to Learn JavaBeans™

(Herndon, VA) - MindQ Publishing Inc. has announced Inside JavaBeans™, the addition to the company's comprehensive suite of Java tutorial curriculum designed to help programmers, application developers, Web site designers and corporations training senior software engineers to quickly learn the cutting-edge Java technologies that use the Internet, Intranet and Extranet as tools for commerce, communication, research and entertainment.

Inside JavaBeans details the component architecture that is the key to programming in Java and working with JavaBeans - how they can be visually manipulated and customized without writing code, wired together to exchange events and data and triggered into by messages fired from



other beans.

Inside JavaBeans features multimedia-driven sound, animation and text. On-line help is available at the click of a button, as is an illustrated, hyperlinked glossary of hundreds of definitions and complete source code listings for all projects.

Inside JavaBeans requires an 80486 or Pentium-based PC; Windows 95 or Windows NT; a SVGA video adapter and monitor; a double-speed or faster CD-ROM, 8MB RAM and 3MB hard-disk storage space. A sound card and speakers are recommended.

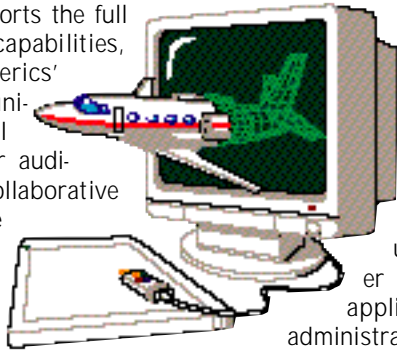
Inside JavaBeans is available immediately for \$59.95. Users may order from MindQ at www.mindq.com or by phone at 800 MIND-008 or 703 708-9380. ☛

Visual Numerics, Inc. Unveils JWAVE™, a Java™-based Visual Data Analysis Tool

(Boulder, CO) - Visual Numerics, Inc. has announced plans to ship JWAVE 1.0 for diverse Intranet and Internet visual data analysis applications. The JWAVE Java components for PV-WAVE® let users create visual and numerical data analysis client applications written in Java. These Java applications communicate with a PV-WAVE server, which acts as a graphics and numerics engine for the client.

JWAVE provides Java classes for plotting data and supports the full range of PV-WAVE capabilities, allowing Visual Numerics' customers to communicate numeric and visual information to a wider audience and facilitate collaborative computing. At the same time, it bars access to critical data.

JWAVE consists of a set of Java components (classes) based on Netscape® IFC, a library of Java classes designed by Netscape Communications Corporation. Part of the Netscape ONE open network environment, IFC enables developers to rapidly construct and deploy highly interactive, network-centric applications. Using JWAVE, application developers can create Java programs that perform complex numerical analyses and generate 2-D plots, surfaces, contours, animations and other graphical data representations. With its common interface, JWAVE supports real-time collaboration, reduces distribution and infrastructure costs and



provides necessary encryption features.

A free evaluation copy of JWAVE can be downloaded from Visual Numerics' Web site at www.vni.com. Its starting cost is \$3,395 (U.S.). For more information, e-mail margaret@boulder.vni.com or call 800 364-8880. ☛

Visigenic Object Request Brokers Ship as Part of Netscape Communicator and Netscape Enterprise Server 3.0

(San Jose, CA) - Visigenic Software, Inc. has announced that its ORBs have shipped as embedded products in Netscape Communicator client software and Netscape Enterprise Server 3.0, Netscape's award-winning Web server software. In addition, Visigenic announced that an upgraded version of VisiBroker for Java is immediately available and includes new features co-developed by Visigenic and Netscape Communications Corporation as part of the Caffeine project announced late last year. The combination of Visigenic's VisiBroker ORB technology and Netscape's client and server software provides an open platform for developers to easily build next generation, multi-tiered distributed applications.

Visigenic's VisiBroker for Java is embedded into Netscape Communicator client software. Netscape Communicator users now have the software infrastructure in place to run Java applets that interoperate with object servers using IIOP, resulting in higher performance Web-enabled applications that require less administration. Both VisiBroker for Java and VisiBroker for C++ have been embedded in Netscape Enterprise Server 3.0, the cornerstone of Netscape SuiteSpot, Netscape's integrated suite of server software. The VisiBroker ORBs enable developers to leverage existing applications based on CORBA to build multi-tiered applications across the Internet and/or enterprise Intranets. In addition to the embedded ORB technology, Netscape developers can take advantage of the single-user VisiBroker development licenses for Java and C++ ORBs that are bundled with Netscape Enterprise Server.

Visigenic's VisiBroker for Java incorpo-

rates Visigenic's Caffeine technology, which makes it easier for developers to build distributed applications encompassing both Java and CORBA. With this latest version of VisiBroker, Visigenic joins Caffeine technology development and the wide availability of Netscape client and server software.

For more information, see Visigenic's Web site at www.visigenic.com, call 415 655-3978 or e-mail stevey@visigenic.com. ☛

Oracle's NCI and Schlumberger to Provide Smart Card Technology for Network Computers

(NY, NY) - NCI has announced that Schlumberger will be the first smart card manufacturer for network computers running NCI's software. The smart cards for network computers, called the NC Card™, will be available through NCI's value added resellers (VARs) and this summer via consumer electronics manufacturers, including RCA and Zenith.

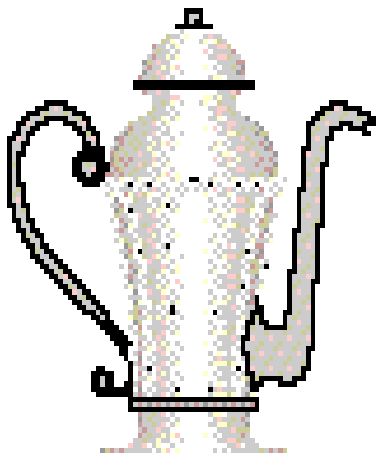
This announcement will enable a new class of users to access information and communications and conduct business anywhere with a network computer. With the NC Card, a person can travel anywhere and access applications and data from the network, wherever there is a network computer.

The NC Card is the user's "network passport." It is a smart card with a small, highly secure, silicon chip that stores information. Like a bank ATM card, it identifies each user to the network so users aren't required to remember complex connect strings, phone numbers and other detailed information. A PIN is used to identify the holder of the card. Upon positive identification, the user can privately access network-based information, databases and services such as electronic mail, Internet applications and electronic commerce.

"The NC Card is a fundamental component of the network computer," said Jerry Baker, CEO of NCI. "The NC Card unlocks the power of the network computer delivering personal information, content and applications to everyone from anywhere in the world. We're pleased to be working with Schlumberger to provide this important technology for our customers."

For more information about NCI, call 415 631-4600 or see their Web site at www.nc.com. For more information about Schlumberger, see their Web site at www.slb.com/et. ☛

Java World Full



JavaSoft Meets Its Karma

THE GRIND

by Joe S. Valley

“Yes, there is some major league bad Karma in the JavaSoft building... and much history of confusion and chaos.” ...

“I remembered that the JavaSoft building has been through many tenants, starting with Apple.”

Joe S. Valley is a scarred veteran of the Silicon Valley wars. It was either writing this column or heading back into therapy. His company can't afford mental health care coverage anymore, so writing is the only option. There are a million stories in the Valley and Joe knows lots of them. Got a good story? E-mail him at Joe@sys-con.com



Joe@sys-con.com

It was a tough day in the Valley. My mouth tasted like 20 miles of bad road, which is what my rear end was feeling as I drove up the winding road to Mt. Hamilton. High above the Santa Clara Valley, I was on a mission to see the Source of All Knowledge. The Chosen One used to be a consultant in Silicon Valley. On one fateful day, after a trip to Club Med, he was trapped in an airliner for 30 hours with 200 French tourists and emerged from the smoke and smell with a new persona, Nostradjava.

I was at a loss to understand the recent product announcements from JavaSoft. My editor was told that Nostradjava could see the future and, thusly, is the only person on the planet who could explain JavaSoft's marketing strategy.

Legend had it he lived in a stone monastery near the mountain peak, subsisting on granola bars and warm Sprite.

As I approached the Mt. Hamilton peak, I was greeted by a creature in long flowing robes, sitting in the front seat of an old Ford van, staring into a brass pot mounted on a tripod.

“Greetings, Old Wise One,” I called.

I heard him mumbling as I continued, “It is said that you see the Valley through the clearest of eyes, from a higher level.”

“You would, too, if you lived on an 8,000 foot high mountain,” he stated.

“So, what happened to the monastery, Oh Keeper of the Eternal Flame?”

“Zoning,” he replied. “Besides, who wants to be in a stone hut when the shaking starts?”

“Earthquakes?” We don't need any more of those. The last Big One took out my entire collection of Elvis beer mugs.

“No, the shaking that will start when the Valley finally understands that Microsoft is winning everything. People must change, and quickly!”

“You mean with innovation, creativity and vision?” I asked.

“No, buildings. Man, there is bad Karma in some of the buildings in the Valley. People need to move to better locations!”

“Wait a minute, why are we talking about buildings? My editor told me to get a prediction from you on the future of JavaSoft. How will they compete with Microsoft? Will there be embedded Java applications? Will there be a Java Network Computer? And...what is the fate of JavaOS? If I don't come back with a story on JavaSoft's strategy, I'll be back to writing for COBOL Weekly!”

He turned from me as his eyes became

glassy. Nostradjava peered into his brass pot and, like his namesake Nostradamus, began speaking in quatrains. Unfortunately, I'm a little rough with my 16th century French, so I asked for a translation.

“If you seek the answers, look to the land,” he mumbled.

“Land, land,” I pondered ...“You mean the JavaSoft building? We are back talking about buildings!”

“Yes, there is some major league bad karma in the JavaSoft building... and much history of confusion and chaos.”

I remembered that the JavaSoft building has been through many tenants, starting with Apple.

He was on a roll as he switched back to his French accent. “It was there that the Little Kingdom of Apple moved its flag at the peak of its power. It was there that the Sculley of Pepsi defeated Steven of Jobs in a legendary battle on the plains of the third floor. It was there that the Huns from the east routed the Cola kingdom and created the disastrous Magna Performa. After the alliance with Blue of Big, the principality of Taligent was formed in that building. Surely, you remember that reign.”

“I sure do.” I was in job search mode then and every time I talked to someone at Taligent, another bright-eyed Yuppie Puppy would ask me why I was good enough to work for them, a company with no leadership, no products and no future.

“So, is JavaSoft doomed by their building? Is the building a locus of bad Karma? If so, why do they have such a nice cafeteria?”

Nostradjava stared toward the North. “To survive, JavaSoft must move closer to the center of the Force, away from the Land of the Sun.”

His Harvard MBA training was evident. I could just picture the size of the consulting bill that my editor was going to get. “So, Oh Great One, where should JavaSoft go to seek their destiny?”

He looked at me and smiled. Switching off his French accent, he said, “Tell your friends at JavaSoft my brother is in commercial real estate. He can get them a great deal on a tilt-up right outside of Redmond, Washington!”

As I headed back down the mountain, I pondered the fate of JavaSoft. Then I heard a commercial on the radio for the latest Apple promotion – buy a t-shirt, get a free Macintosh. And, if you act now, they will throw in a free Taligent coffee mug.

It was a long drive home. ●

Microsoft Ad

KL Group Full Page Ad