SYS-CON

NEW LOOK!
Our Name has Changed!
FORMERLY LinuxWorld MAGAZINE

ENTERPRISE
OpenSource
MAGAZINE

Visit us at **opensource.sys-con.com**

THE LEADING MAGAZINE FOR ENTERPRISE AND IT MANAGEMENT

AUGUST 2006 VOLUME 4 ISSUE 8

LINUX
AUTHENTICATION USING PAM AND LDAP

+PLUS

Making Sense of Virtualization
The Tidal Wave of Innovation Has Begun

Testing
Using Kernel Modules To Help Test Applications

Ajax
AJAX and the Maturation of Web Development

What's Inside...

Linux Authentication Using PAM and LDAP
by Quan Dinh and Clayton Donley

[16]

Making Sense of Virtualization
by Alex Vasilevsky

[20]

OpenVista at Midland Memorial Hospital
by Frank Pecaitis

Open Source Is Not a Trend. It's a Paradigm Shift

Why free and open source is more than a fad

By Mark R. Hinkle

I recently read an article in the "mainstream" media that gave me pause. The author made an assertion that the current trend towards Open Source might just be a passing fad. I thought about this and looked critically at the software industry, thinking about whether there was merit in that statement. After all, we have seen plenty of high flyers peter out in a software industry riddled with buzzwords and acronyms-of-the-day. I just don't believe that open source is one of them.

When I worked in the Internet Service Provider (ISP) business, we saw every new consumer Internet fad firsthand. I remember PointCast, a push technology with a screen saver and news ticker that sucked up bandwidth and reduced office LANs to a crawl. I recall the days when every other technical support call was for help setting up the hot new Web browser, Netscape (the code that has, ironically, been given a second life as the basis of the free and open source Firefox), or the time when when thin clients were first championed by Sun Microsystems. However, not everyone grasped the idea that we would have always-on "dumb terminals" and that all our applications would live on the servers carefully maintained and backed up by application service providers (ASPs), though this vision may someday be realized by Google (who may be setting the stage with their Google Desktop (http://desktop.google.com). Thin clients seemed to have found ways to remotely access internal applications in order to keep data from leaving secure networks. Besides, like so many others, I still love my data-laden, fat-client laptop, and as much as I enjoy tools like hosted e-mail and online back-ups, I don't foresee parting ways with my laptop anytime soon.

What makes free and open source software more than a fad? The notion of free software is nothing new - in fact, it's how software was originally distributed. In the earliest days of computing, you usually got the source code for the operating system at the time you purchased the mainframe. At that time, of course, there were very few computers - most of which were for use at large companies, government agencies, and universities - so users often shared programs and source code. Since free software has been around for over 40 years, it can hardly be called a fad. Furthermore, there are more and more long-term projects that continue to gain momentum. Linus Torvalds started Linux, for example, in 1991, and 15 years later, the operating system is enjoying quarter after quarter of double-digit growth in the server market. That's hardly a flash in the pan. Apache, too, the dominant Internet Web server, has been growing by 4.4% since July, and holds 63.09 % of the web server market, according to Internet services company Netcraft (http://news.netcraft.com/archives/web_server_survey.html).

Not only do open source markets have a reliable history, but their future, too, is nearly guaranteed by the continued need for the adoption of open source software in order to sustain the efficiency they have brought to the industry. First, open source software is by and large a pull model. Before any corporate vendor can get involved, software is usually developed to solve specific problems faced by end users who they self-identify in market. Most people like to showcase their

About the Author
Mark R. Hinkle is the vice president of strategy and corporate development for Emu Software as well as the editor-in-chief of **Enterprise OpenSource Magazine**. *Mark served on the Formation Board of The Desktop Linux Consortium and is also the author of* Linux Business Desktop Migration for Windows Users *from Charles River Press.*
mrhinkle@encoreopus.com

Reclaim lost time

The world's first Linux management appliance

The Intrepid M™ appliance performs the most important Linux management tasks in a fraction of the time you now spend. You'll save both time and money with a single point of control.

The Intrepid M does these tasks and more:

- **Quickly provision servers** — in minutes—even from bare metal

- **Easily deploy software and patches** – eliminating lengthy installations

- **Efficiently migrate software** — from one piece of hardware to another

- **Instantly rollback** — undo changes in minutes, even to the distant past

"Levanta reduced an eight-hour process down to five minutes."

-Arty Ecock, Manager of VM Enterprise Systems, City University of New York (CUNY), Computing and Information Systems

Levanta Intrepid M™

Download the Illuminata Analyst Report

Levanta Marries Provisioning with Data Virtualization

www.levanta.com/lwm0406

Find out why the Intrepid M is the cutting edge in Linux server management.

LEVANTA®
www.levanta.com
1.877.LEVANTA

CEO Vision of Linux

Open Source returns power to the customer

By Ron Hovsepian

It's important not to get too carried away with "the latest tech trend." Technology changes more rapidly than any other sector, and this year's "must have" technology is quickly made obsolete, or so it seems. However, every so often something significant comes along that truly changes the game. Mainframes yielded to client/server, which in turn was replaced by the Web as the dominant computing paradigm. I believe Linux and Open Source more broadly represent a similar game-changing force.

Open Source does two things that fundamentally differ from how proprietary software has worked over the last two decades. First, Open Source returns power to the customer. In the past, customers were largely at the mercy of vendors when considering their computing options. Yes, you could choose between Oracle and SAP, but the vendors determined the timing, features, and price of what you got. If you needed something quickly to meet a business need, you had to build it yourself, or lobby the vendor to include the feature in the next release, which might be years out. With Open Source, things move very quickly. The Linux kernel is updated regularly. New applications can be developed and deployed in weeks, rather than months or years. Customers can use their own resources, but also tap the resources of the Open Source community to get features they need. The flexibility of Open Source gives control to the customer.

The second major difference with Open Source is that it delivers affordable innovation. In the proprietary model, vendors would control what went into their products, spend mounds of money innovating in-house, and charge customers a premium for getting "bleeding-edge" technology. With Open Source, innovation happens from the ground up, includes input from both independent and corporate software developers, and is available, for free, for companies that want to use it. There's no formal product requirements document, no product launch timetable. Developers create solutions to fix holes they see. If the fix is good, it gets adopted. If not, something better is invented. It's an informal process, but it works. And not only is there no premium for this innovation, it's free for the taking.

Linux now provides a viable, enterprise-level platform that works from the desktop to the data center. Novell offers solutions across this range, but we didn't invent Linux and we certainly don't own it. We're a constructive member of the community that has made Linux into an enterprise-quality platform. Our contributions have included applications security technology, management tools, and, on the desktop, we've worked on OpenOffice. Innovation at the server and desktop level is now surpassing what's happening in the proprietary world. The number of applications that are moving to Open Source is expanding daily. Today, Novell has some 2,000 applications certified for our SUSE Linux Enterprise platform.

This growth will only continue. Open Source is a circle. The more solutions that are built for Open Source, the more Open Source is adopted, the more solutions need to be built. Open Source is also a great beneficiary of network effects: the more people use it, the more valuable it becomes because more resources are devoted to improving it. There's no doubt in my mind that Open Source will become the dominant approach to software development in the future. Proprietary solutions will always be there. We see a world with a mix of Open Source and open standards-based solutions. With open standards, integrated applications, and solutions built on a service-oriented architecture and Web Services, plus a never ending well of innovative Open Source solutions, the customer wins. We're proud to be part of this trend and are working to assure it's sustainable. ⭕
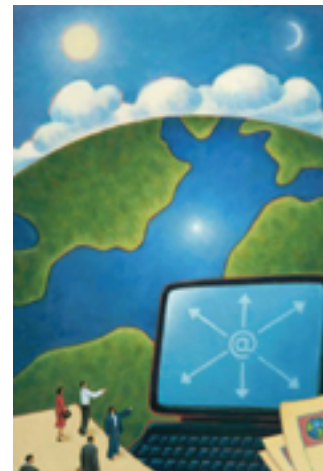
About the Author
Ron Hovsepian is president and CEO of Novell.

Electronic Communications Retention, Retrieval & Supervision

Open Source/Open Source-based versus Commercial Solutions

by Arthur Riel

The market for electronic communications retention, retrieval, and supervisory systems is growing at a rapid pace. This growth is driven by a number of factors including the need for better regulatory control over corporate communications, Sarbanes-Oxley compliance, enhanced electronic discovery tools for litigation support, better control over internal policies, the mining of critical information from the unstructured data that is electronic communications, and enhanced mailbox management. Open Source and Open Source-based solutions offer clients distinct advantages over proprietary and/or 100% commercial solutions.

About the Author

Arthur Riel, the CTO of Lighthouse Global Technologies, is a world-renowned expert in the field of object-oriented design/development, with over 20 years' experience working with dozens of companies around the world on designing, developing, and managing large-scale software engineering projects. Arthur is frequently invited to lecture at conferences as a guru in process control, code generation, telephony, straight-through processing, and financial services. For eight years prior to joining Lighthouse, he worked for several financial firms, including Greenwich Capital Markets, Goldman Sachs, Merrill Lynch, and Morgan Stanley.

Arthur.Riel@lighthousegt.com

Electronic Communication Archiving Market Segments

The market for electronic communications retention can be divided into six key areas: regulatory compliance, Sarbanes-Oxley compliance, litigation support, internal policy compliance, knowledge management, and mailbox management.

The regulatory compliance facet ensures that all laws and regulations from government entities such as the SEC, NASD, and NYSE are followed. The largest clients in this space include banks (investment, commercial, and retail), other financial institutions, and insurance companies. Large institutions have spent considerable resources complying with regulations over the past five years and have, for the most part, been fined into submission. The most active area of this market segment is small banks, credit unions, and (in particular) hedge funds.

Small banks and credit unions have flown under the regulatory radar with respect to government agencies such as the SEC. Many are aggressively following in the footsteps of their larger brethren and investing their resources in the area of regulatory compliance. Hedge funds are technically unregulated entities with respect to electronic communications retention, although there has been considerable saber rattling from the SEC and other regulatory bodies over the need to police these entities better. As such, our industry is attracting considerable interest from hedge funds with respect to archiving. It's unclear whether this is in response to vocal government regulators or to support electronic discovery in potential future litigation.

A market sub-segment of regulatory compliance is found among those clients for whom compliance with the Sarbanes-Oxley Act requires that they retain their electronic business communications. While Sarbanes-Oxley requires the retention of business records by all public companies, this segment is only beginning to be interested in archiving products. Surely, most public companies have personnel planning for formal document and electronic communications archiving. They are canvassing the existing technologies and putting document-retention policies in writing. However, they are not purchasing products in this space at a high level. I predict that in the near to medium future one of the larger mid-sized company will run afoul of Sarbanes-Oxley because of financial shenanigans or high-profile litigation in which they can't produce the appropriate documents. The SEC will extract a high fine from the offending company, causing many small and medium-sized public companies to fulfill their obligations, resulting in a spike of acquisition in this technology area. At this

stage, the regulatory threat doesn't seem real, especially for companies at the smaller end of the public entity universe.

In Europe a similar view exists around Basel II, a set of regulations that seeks to guarantee that all public companies have adequate risk analyses, mitigation strategies, and financial reserves. Basel II is much larger in scope than Sarbanes-Oxley in the United States, since it deals with all risks as opposed to the financial risk areas covered by Sarbanes-Oxley. Basel II is viewed by many public companies as more theoretical than practical, with many companies postponing all but the most rudimentary planning exercises. Just as with Sarbanes-Oxley, the smaller the company the less chance they're adequately planning for the regulations.

The marketing strategy for both Sarbanes-Oxley and Basel II is to ensure that potential clients know you exist and understand your technology, assist them in their planning exercises for the record retention portion of their compliance programs, and be ready to provide solutions quickly when they're motivated by regulators to resolve their compliance issues in this space.

An important driver in electronic communications retention is for the support of electronic discovery in civil litigation. With the estimate that over 90% of the companies with over 100 employees will be involved in civil litigation in the next three years, almost no company can ignore the role of document archiving. Judges have shown no patience with civil litigants who can't produce their documents, including electronic communications, in a timely fashion. Sanctions as well as partial or full default judgments aren't unusual when the material isn't forthcoming. Companies that attempt to produce deliverables such as e-mail that matches certain sender/receiver and keyword criteria from backup media are often surprised at the amount of time and expense third-party companies charge. In-house attempts to extract the material aren't typically cheaper and almost always take much longer. In fact, the cost of electronic discovery in such situations often leads to a disadvantageous settlement in spite of the merits of the case.

Through the use of an electronic communications archive, electronic discovery costs are less than 10% of the cost of the backup media, with delivery time reduced to minutes or less. For companies motivated by this facet of archiving, the cost of maintaining the archive is viewed as insurance. Given the litigation statistics, this is an insurance policy that will pay off quickly for most companies.

Companies that archive their electronic communications in an easy-to-search repository have a convenient central location for enforcing their internal compliance policies. Whether searching for keywords or phrases that are deemed inappropriate, or enforcing a Chinese wall (e.g., e-mails between investment bankers and equity research analysts are prohibited), the archive can be used to search for violations. Once violations have been uncovered then corrective action can take place. While retention systems can be used to find policy violations after the fact, companies that require blocking e-mails before they're sent will want to explore the area of e-mail pre-compliance tools.

Many companies that have bought e-mail archive products quickly realize that there's considerable value in the unstructured data. Most communications between a company and its clients are captured, as well as the thought processes that led to strategic decisions. The problem is sifting quality information from the sea of unstructured data. Products that offer advanced reporting capabilities and/or data mining modules are especially important here. A key driver is the use of an e-mail archive to research effective sales and client support. Senior managers realize there's a problem between a salesperson and/or force and its clients when a client or clients is lost. These managers are using their e-mail archives to look for systemic problems that can be rectified through better client education and problems with a particular salesperson and/or client relationship. Companies using the valuable knowledge contained in their e-mail archives can be more proactive and competitive than those that don't.

The last market segment is the use of an e-mail archive to get better and lower-cost mailbox management. The combination of employees keeping more of their e-mails on their mail servers and the rapidly increasing average size of each e-mail is causing most firms to frequently upgrade their e-mail plants. These upgrades can take the form of adding storage devices/arrays or adding more e-mail servers. In either case, the cost to the firm is high due to the need for high-availability (i.e. expensive) storage indigenous to e-mail plants. Through the use of an e-mail archive, companies can impose strict mailbox quotas on the mail servers, requiring users to go to the e-mail archive for older email. E-mail archives make use of cheap, deep storage solutions due to the lack of high-availability requirements like those of mail server plants. E-mail archiving products that integrate mailbox management features are almost guaranteed to pay for themselves

quickly in reduced e-mail plant upgrades. For that reason, most companies looking to acquire e-mail archives for regulatory purposes are also looking to capitalize on the mailbox management facet to enrich the ROI of the technology.

Open Source/Open Source-based versus Commercial Archive Products

There are a large number of electronic communication retention, retrieval, and supervisory systems on the market today. Companies seeking to acquire technology in this space must first examine their requirements. These requirements must then be cross-referenced with the attributes of each product under consideration. Requirements collecting examines the needs of the company across a number of criteria:

- What is the primary purpose of the technology acquisition, e.g., regulatory requirements, litigation support, mailbox management, internal policy compliance, knowledge management? Often more than one purpose must be accommodated.
- What types of electronic communications is the firm attempting to capture, retain, retrieve, and/or supervise? Is it just e-mail or is instant messaging, faxes, and/or voicemail included?
- What kinds of e-mail is the firm using? Is it Microsoft Exchange mail, or are other e-mail servers in use? Many products work only with Exchange journal files, rendering them useless for companies with other server technologies such as Lotus, gmail or IMail.
- What are the firm's infrastructure requirements? Is it willing to acquire infrastructure such as servers or storage to accommodate a product, or does the product need to fit into the existing infrastructure?
- What is the firm willing to spend to acquire technology in this area? Total cost includes not only software licenses but any necessary hardware, third-party software, installation and configuration charges, and ongoing maintenance and support.

One key attribute to research when selecting a solution in this area is the background of the company offering the solution. It's important to pick a company whose senior management and/or IT architects have extensive experience in compliance technology and the regulations that make up the field. There are many companies that simply reacted to the analysts' predictions that the electronic communication compliance market would have double-digit growth
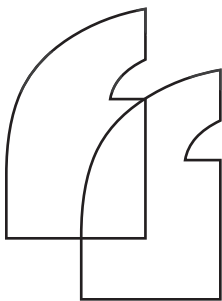
I predict that in the near to medium future one of the larger mid-sized company will run afoul of Sarbanes-Oxley because of financial shenanigans or high-profile litigation **in which they can't produce the appropriate documents**

for the next 10 years and quickly developed a product for the marketplace. These companies developed a set of requirements based on what they thought the market, and its regulators, required. They lack the practical experience necessary to develop a solution that captures the nuisances of regulations.

For example, does the product do keyword searches across all attachments including a re-cursive descent into composite file types such as zip and rar? It had better if you're responding to the SEC, or a judge in a civil litigation. Does the solution offer quick and easy methods for running a number of inquiries, in parallel, with only one data entry requirement? Most inqui-ries want the same parameters (date range, keyword search) run over a set of a dozen or more people. Does the solution offer bcc injection (i.e., the optional annotation of an e-mail header statement noting any recipients that were bcc'd). This is helpful in reminding regulators why that particular e-mail is present in the results, or they may think you're burying them in useless data. Does the solution offer handy refinements such as filtering the com-munications for internal-only or external-only communications? Does it let the user stipulate that all receivers must be on each e-mail in the set to reduce the volume of communications for review? All of these and more are obvious to anyone who has worked extensively with regulators, lawyers, and/or judges. And they're left out of many products developed by less experienced designers.

Electronic communication retention, re-trieval, and supervisory products that are Open Source or Open Source-based offer a number of advantages over many commercial products. The greatest advantage that clients get with Open Source applications is the fact that they tend to be infrastructure-agnostic. Most prod-ucts tend to be written in Java to accommo-date a number of operating systems including Linux, Unix (e.g., AIX, Solaris), and Windows. Those that use a database as part of their implementation typical do so via a standard API such as JDBC simplifying portability to a

number of common databases including DB2, Oracle, Sybase, MySQL, and SQL Server. Those products that index keywords do so via open sourced solutions such as the Lucene indexing framework. The client is given a wide variety of choices in storage solution including IDE, SCSI, SAN, and NAS storage solutions. Commercial products are often limiting in their choice of infrastructure with many companies creating alliances with a particular storage vendor.

By being infrastructure-agnostic, Open Source-based products let companies save money by avoiding unnecessary hardware purchases (such as product-specific storage or servers) as well as third-party software license needs (such as specific database products, indexing tools, and reporting tools). While licenses for truly Open Source products are free, companies will often have to pay for professional services to install and configure the products. Many companies are also wary of products without a formal support/mainte-nance contract. Third-party companies often offer such contracts much the way companies like Red Hat offer support for Linux. Open Source-based products offer a commercial product that exploits Open Source software for third-party needs such as databases, operating systems, reporting, data mining, and indexing. These products charge a nominal fee that is often much less than competing commercial products, while offering a formal support structure for clients. This can provide custom-ers with a high level of IT value, minimizing costs and risks optimally.

In the end, companies must marry their particular requirement sets with the best available product in the marketplace thereby minimizing their firm's risks. They must then do so at a price point that meets their financial requirements, keeping in mind that license cost is only one component of the total cost of ownership. Open Source-based solutions can provide many users with a high IT-Value proposition, providing optimal cost benefit tradeoffs while minimizing compli-ance and operational risks. ⬤

The Power of Being There.®
Times two.

Discover how the combined power of Avocent and Cyclades IT infrastructure management solutions can take you and your data center to the next level. KVM, serial and power – all over IP. Plus, Intelligent Platform Management Interface (IPMI) and embedded KVM. The Power of Being There®. **Times two.**

Avocent®

cyclades

An Avocent® Company

Visit www.avocent.com/powerx2

More Than Webbed Feet

The Internet, Linux style

by Rickford Grant

These days, average home computer users spend more time surfing the Web and writing e-mail messages than doing just about anything else. Even if you're not much of a surfer, there are still numerous other applications that aren't really Internet applications per se but that still make use of the Internet in some way, such as gathering song and album information when you rip audio CDs to create MP3 files. Having a computer that isn't hooked up to the Internet is like buying a new Maserati and then refusing to take it out of the garage.
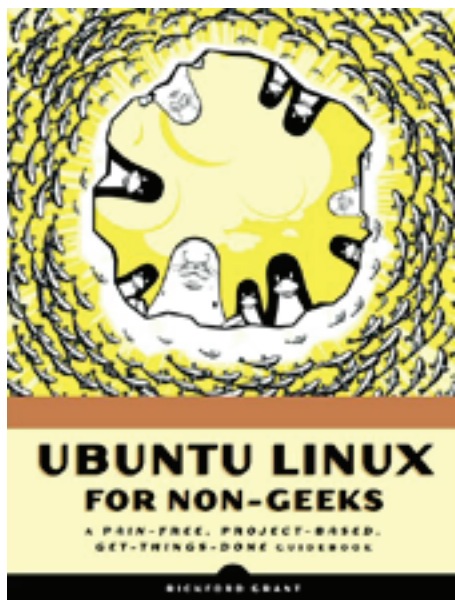
About the Author

Rickford Grant, author of Linux for Non-Geeks and Linux Made Easy (both No Starch Press), has been a computer operating system maniac for more than 20 years. From the Atari XL600 to today's Linux machines, he has been the guy behind the help desk for family, friends, and colleagues. Rickford currently resides in Raritan, New Jersey, where he spends his working hours as a teacher and his free time cycling along the Delaware and Raritan Canal or annoying his neighbors with his Nyckelharpa (a Swedish key fiddle).

email?

Of course, how you connect to the Internet depends on your hardware and provider. There are a number of possibilities in this area, including high-speed local area networks (LANs), cable modems, and ADSL connections from phone companies. Most computers also have an internal 56Kb/s modem or can be connected to external dial-up modems for slower connections over regular phone lines. Depending on what you've got, setting things up on your system should prove a cinch in the case of LAN connections and any others that make use of your Ethernet port (such as cable modems), possibly a bit more work in the case of wireless connections, and sometimes a bit of a challenge when it comes to the ol' dial-up connections. In this excerpt, you will learn how to set up these connections and learn a bit about what Linux has to offer in terms of the most commonly used Internet applications – your Web browser and e-mail client.

Firefox: Your Internet Browser

Now that you are connected to the Internet, you no doubt want to get down to some cyberspace discovery and exploration, and the most commonly used means of doing that is with a Web browser. The default Web browser in your Ubuntu system is Firefox, which is enjoying increasing popularity in not only the Linux world, but in the Windows and Mac worlds as well. Chances are you are already a Firefox user, but if you are not, then you needn't worry – things work more or less the same in all browsers. That being the case, you should be able to use Firefox's basic features without any instruction. Of course, there are some features that do distinguish Firefox from its competition, so I will mention those.

Controlling Browser Window Clutter with Tabs

Usually when you click a link on a Web page, the new page opens in the same window. On some pages, links are coded so that the new page opens in a new, separate window, or maybe you occasionally opt for opening a link in a new window by right-clicking the link and then selecting the Open Link in New Window option. This can be very useful; however, once you have more than a few browser windows open, it gets sort of hard to find what you're looking for in all those open windows. It can also slow things down a bit.

This is where Firefox's tab feature comes in handy. To see how it works, try it out yourself right here and now. Open your Firefox browser by clicking the launcher on the top GNOME Panel (or going to Applications Internet Firefox Web Browser); then Google the word nyckelharpa using Firefox's handy search box, which is next to the word Go at the top-right corner of the browser window (see Figure 1). By default, Firefox will perform searches for keywords entered in the search box using Google. You can, if you like, select other search engines by clicking the G icon in the search box and then making your selection. Amazon.com, eBay, and Yahoo! are available, to name a few, and you can even add others. For now, however, let's stick to Google for our present search by typing nyckelharpa in that search box. Once you've finished typing, press the ENTER key, after which a page of Google results should appear in the main page of the Firefox window.

Note: While tabbed browsing is no longer as unique as it once was (Safari has the feature built in, though you have to enable it yourself, and Internet Explorer now has the same functionality available as a downloadable add-on), it is implemented and enabled by default in Firefox.
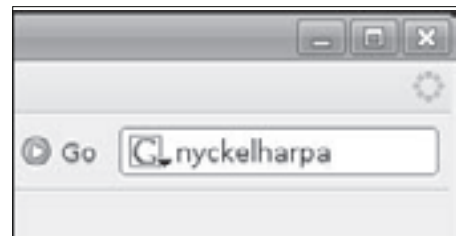
Figure 1: Performing a Google search from the Firefox search box
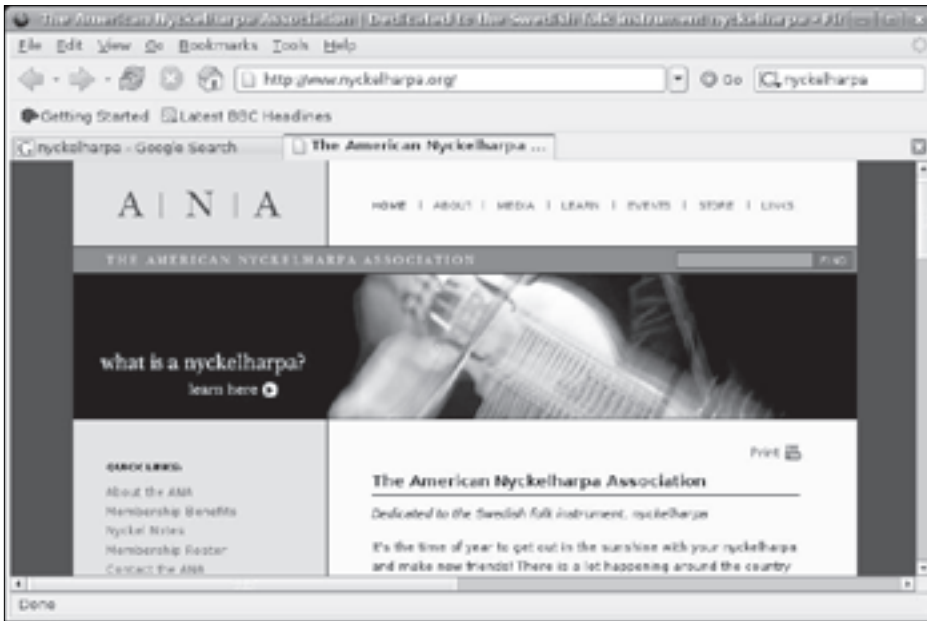
Figure 2: A link opened in a new tab in Firefox

Figure 4: Seeing what extensions you have and getting more in Firefox

Figure 5: Firefox seeks your permission before installing an extension

The top result should be the American Nyckelharpa Association, and you are now going to open that page in a new tab, rather than in the same or a new window. To do that, right-click the link, and in the popup menu that appears select Open Link in New Tab. You can, if you prefer, make things a tad easier and dispense with the popup menu selection step by simply clicking the link with both mouse buttons simultaneously or by holding down the CTRL key as you click the link. Either way, the new page will appear in a new tab, while your original page of search results remains, ready and waiting in the other tab (see Figure 2). I am pretty confident in saying that, once you get used to this feature, you will wonder how you ever got along without it.

Other Firefox Features: Popup Manager

Firefox has a number of other useful features. One is its Popup Manager, which suppresses those annoying popup windows that often appear when you access a new Web page. You can enable or disable this feature from the Preferences window (Edit Preferences) by clicking the Web Features icon in the left pane of that window and then checking or unchecking the box next to the words Block Popup Windows. You can also permit certain sites to provide popup windows (some popups are not only useful, but necessary for the correct functioning of a site) by clicking the Allowed Sites button next to that Block Popup Windows entry and inputting the URL for the site in question.

One of the coolest things about Firefox is that it allows you to further expand its functionality by adding various extensions. These extensions include all sort of things; many are quite func-

tional, while others are just plain fun and goofy. They range from blog-writing tools to image viewers. For this project, however, we will be installing a cool weather station of sorts, called Forecastfox, that allows you to view not only the current weather conditions in your area (or any other area of your choosing), but also a two-day local forecast, Doppler radar maps, and more – all from AccuWeather.com. All of this is available at the click of a button from the Status or Menu bars, or the Bookmarks or Navigation toolbars – your choice (see Figure 3).

Downloading and Installing the Forecastfox Extension

To get started with the process of installing any Firefox extension, you have to first, quite logically, find and download one. To do this, go to the Firefox Tools menu, and select Extensions. The Extensions window will then appear (see Figure 4), showing you the extensions you already have installed, which at this point should be only one. To add more, click the Get More Extensions link at the bottom-right corner of the window, which will bring your browser to the Firefox Add-ons site.

At this point, you would normally browse for things that seem of interest to you; check to make sure they don't have any special requirements (such as Microsoft Windows – a few do), and then download and install the

Figure 3: The Forecastfox extension installed in Firefox
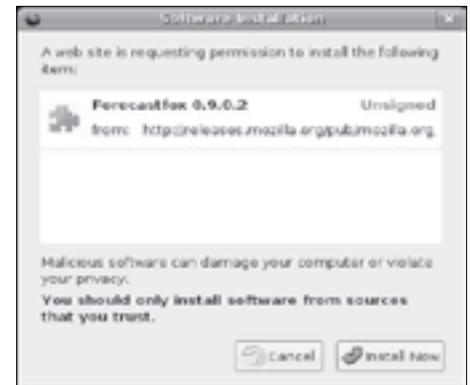
extension that has struck your fancy. In this case, we already know what we are going to install, so I'll just tell you where to find it.

To find the Forecastfox extension, first try having a look at the Top Downloads section of the page, as it is often there. Otherwise do a search for it in the search box on the Extensions page. You might also just type the URL where it is currently residing: https://addons.mozilla.org/firefox/398. Once you've found it, click the Install Now link for that item and then wait. Sometimes the download will take a bit of time, sometimes less. Either way, just be patient, and don't keep clicking the link. The extension will be in the process of downloading even if seems as if nothing is happening, and when it is done, a window will appear telling you so (see Figure 5).

Once the window appears, all you have to do is click the Install Now button. The Extensions window (previously seen in Figure 4) will then pop to the front of the currently open Firefox window and indicate the progress of the download. When the download is done, the new extension will be added to the list of extensions with a message telling you that Forecastfox will actually be installed once you restart Firefox. The implicit command thus being: restart Firefox.

Setting Up the Forecastfox Extension

When Firefox first starts up after you've installed the Forecastfox extension, you will see the Forecastfox Options window (see Figure 6). In that window, type your zip or postal code (or that of any other area for which you want weather information) in the text box next to the word Code. If you're not sure what the zip or postal code for your desired locale is, click the Find Code button to select that location by name.
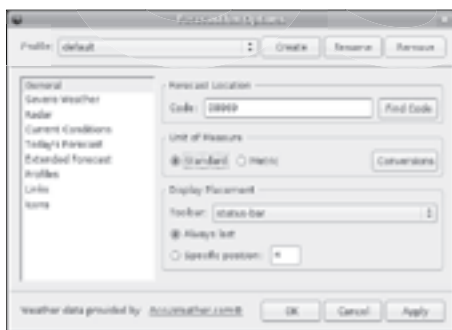
Figure 6: The Forecastfox Options window

In the Forecastfox Options window, you can also choose whether you want the temperatures shown in Fahrenheit or Celsius and where you want the information to appear in your browser window. The Status bar at the bottom of the browser window seems to be the least intrusive location, so that is what I've chosen. Of course, you are free to place it where you like. When you're done, click the Apply button and then the OK button, and you'll be on your way to intense meteorological entertainment.

E-mail with Evolution

Evolution is the default e-mail program in Ubuntu, and it could probably best be described as a better-groomed, spunkier clone of Microsoft Outlook (see Figure 7). It allows you to send and receive mail, make appointments, and keep a list of tasks. It can also filter junk mail, which is a necessity these days, and even synchronize with your PalmPilot, if you still have one of those. Also, if such things are important to you, it is a much more handsome program to look at than Outlook.

To use Evolution, just click the e-mail launcher on the top panel or go to the Applications menu and select Internet Evolution Mail. When you first run Evolution, you will be greeted by a setup wizard, so have the account details you received from your Internet service provider handy. These should consist of your

POP host address for receiving mail, your SMTP host address for sending mail, and your mail password, which is very often different from your Internet logon password. Your mail password is not actually entered during the various wizard steps, so check the Remember this password checkbox when filling in the POP details. When you first connect to your mail server, you will be prompted for your mail password, so you can type it at that time, and you won't have to deal with it again.

An E-mail Alternative: Thunderbird

Evolution is, without a doubt, the most popular e-mail software in the Linux world, but despite its obvious attractions and popularity, I have to say that I don't much like it, though I can't quite put a finger on the reason why. I just prefer the more straightforward Thunderbird for my e-mail chores. In contrast to the multifunctional Evolution, Thunderbird (see Figure 8) is a more mail-oriented program that is very straightforward to use, yet includes most of the most important e-mail functions you've come to expect, such as junk mail filters. In fact, it is remarkably similar to Outlook Express in terms of appearance and handling. It also lacks the quirkiness that always seems to squirrel its way into Evolution in some form or another. The fact that Thunderbird is also available in both Mac and Windows versions means that you may already be familiar with it, or prepared to deal with it if you find it in use on another system you happen to be using.

Thunderbird does not come bundled with Ubuntu, so if you would like to try it out, you will have to download it and install it yourself. Now that you have set up your machine to connect to the Internet, however, you can easily do this after going over the contents of Chapter 5 (okay, so I'm jumping the gun a bit again). Just do a Synaptic search for thunderbird, and then mark mozilla-thunderbird for installation. Once it is installed, you can

then run it from the Applications menu by selecting Internet Thunderbird Mail Client. As I mentioned, both Evolution and Thunderbird are equally capable and possess essentially the same features in terms of mail handling. The difference is primarily a look-and-feel matter. Why not try both Evolution and Thunderbird and see which you like better?

By the way, if you find that you prefer Thunderbird to Evolution, you can add a panel launcher for it so as to make things easier on yourself when you want to run the program. Just go to Applications Internet Thunderbird Mail Client, right-click that entry, and then in the popup menu that appears, select Add this launcher to the panel. You can then remove the Evolution launcher, if you are so inclined.

Other Internet Applications

What I've covered thus far in terms of Internet applications is just the tip of the iceberg (might as well use that worn-out phrase before there aren't any icebergs left, right?). Ubuntu also comes with a couple of other Internet applications that you might want to consider. These include the Instant Messenger client called Gaim (Applications Internet Gaim Internet Messenger), which allows you to use any one of your MSN/Windows Instant Messenger, Yahoo! Messenger, AOL Instant Messenger (AIM), ICQ, Gadu-Gadu, Napster, GroupWise, IRC, or Jabber accounts . . . or all of them simultaneously. If you want to give Internet telephony a try, Ekiga Softphone (Applications Internet Ekiga Softphone) also comes bundled with your distribution, so you need not despair.

There are still more Internet applications that you might want to consider downloading and installing, such as WiFi Radar and Thunderbird, but there are a host of others. If this all sounds pretty enticing to you, get those fingers of yours flipping and move on to the next chapter—the mother lode awaits!
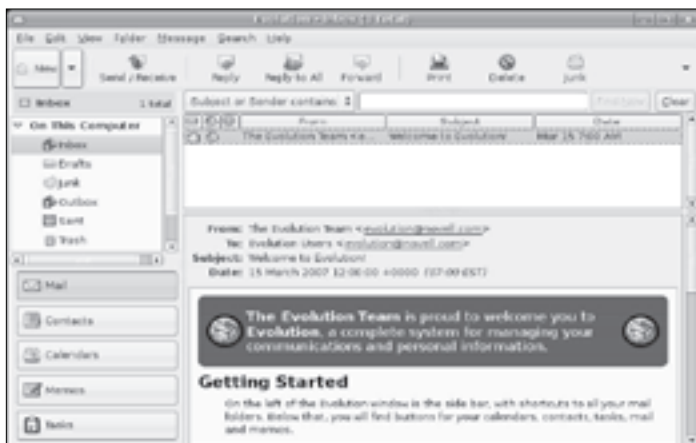
Figure 7: Ubuntu's default e-mail client - Evolution
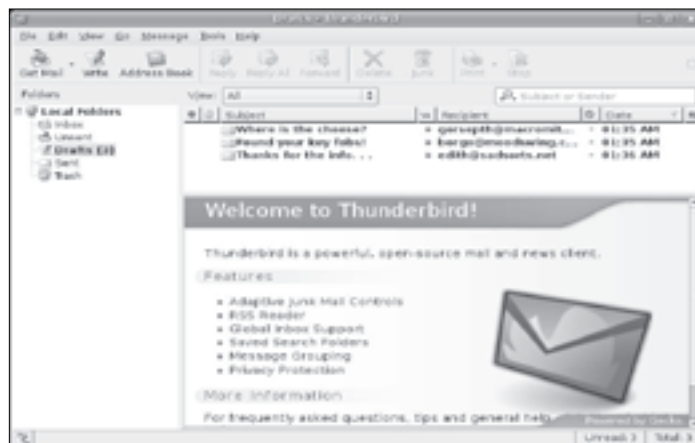
Figure 8: The Thunderbird email client

Making Sense of Virtualization

The tidal wave of innovation has begun

by Alex Vasilevsky

About the Author

Alex Vasilevsky brings over 20 years of extensive engineering, technology leadership, and management experience to Xen-based solution provider, Virtual Iron Software. As a co-founder, he has been instrumental in defining and creating its technology and architecture. Prior to Virtual Iron, he was CTO of Ucentric Systems (acquired by Motorola), a provider of home media software for media centers. He is listed in The History of the Development of Parallel Computing, and won three IEEE Gordon Bell Awards for practical applications of parallel processing research. He has a BS in computer engineering from Syracuse University and a Master's in computer science from Boston University.

alex@virtualiron.com

Companies are finding it increasingly difficult to manage their enterprise data centers as they become highly complex, expensive to build out, and difficult to reconfigure as needs change. In an effort to address these challenges, many IT professionals are turning to virtualization technologies.

Virtualization addresses a number of these issues and offers a variety of benefits including hardware utilization, operational efficiency, and data center agility. However, many customers and their technology partners are becoming increasingly frustrated with the proprietary and expensive nature of the available virtualization software solutions. Luckily, a new wave of virtualization-related technologies is emerging to address these challenges and improve the economics of virtualization.

These emerging solutions are enabling a more dynamic IT infrastructure that helps transform the static, hard-wired data center into a software-based dynamic pool of shared computing resources. They provide simplified management of industry-standard hardware and enable today's business applications to run on virtual infrastructure without modification. Using centralized policy-based management to automate resource and workload management, the solutions deliver "capacity on demand" with high availability built in.

Virtualization 101

Regardless of the increased need for and the constant discussion in the industry around virtualization, many IT professionals are still having difficulty grasping the terminology and comprehending the many choices of hypervisors and hardware that make up the complicated virtualization landscape.

Originally part of mainframe technology, virtualization isn't a new concept. It's been applied to various technology problems throughout computing history and is now receiving renewed interest as an approach for managing standardized (x86) servers, racks, and blade systems.

Virtualization lets administrators focus on service delivery by abstracting hardware and removing physical resource management. It decouples applications and data from the functional details of the physical systems, increasing the flexibility with which the workloads and data can be matched with physical resources. This enables administrators to develop business-driven policies for delivering resources based on priority, cost, and service-level requirements. It also enables them to upgrade underlying hardware without having to reinstall and reconfigure the virtual servers, making environments more resilient to failures.

At the core of most virtualization software solutions is a "virtual machine monitor" or "hypervisor" as it's sometimes called. A hypervisor is a very low-level virtualization program that lets multiple operating systems – either different operating systems or multiple instances of the same operating system –share a single hardware processor. A hypervisor is designed for a particular processor architecture such as an x86 processor. Each operating system appears to have the processor, memory, and other resources all to itself. However, the hypervisor actually controls the real processor and its resources, allocating what's needed to each operating system in turn. Because an operating system is often used to run a particular application or set of applications in a dedicated hardware server, the use of a hypervisor can make it possible to run multiple operating systems (and their applications) on a single server, reducing overall hardware costs.

Server Virtualization versus Data Center Virtualization

Server virtualization is the masking of server resources from server users. The technology can be viewed as part of an overall virtualization trend in enterprise IT that includes storage virtualization, network virtualization, and workload management. This trend is one component in the development of autonomic computing, in which the server environment will be able to manage itself based on perceived activity. Server virtualization is also seen as a likely requirement for both utility computing in which computer processing power is seen as a utility that clients can pay for as needed, and grid computing in which an array of computer processing resources, often in a distributed network, are used for a single application.

While first-generation technologies were limited to working on a single machine or with small clusters of machines, data center virtualization manages the utilization and sharing of many machines and devices including server, storage, and network resources. This enables enterprises to automate numerous time-intensive manual tasks such as provisioning new servers, moving capacity to handle increased workloads, and responding to availability issues. In this environment, any application can run on any machine or be moved to any other machine without disrupting the application or requiring time-consuming SAN or network configuration changes. With these capabilities companies can transform the data center into a manageable and dynamic pool of shared computing resources, enabling IT to rapidly respond to changing business demands and dramatically reduce the costs of managing and operating the data center.

What About Xen?

Xen is a new Open Source hypervisor that is quickly being embraced as an industry standard. It supports the execution of multiple guest operating systems with very efficient levels of performance and resource isolation. Xen lets different operating systems such as Windows and Linux share the same server, and lets development and test systems run at the same time on the same hardware. It has a broad ecosystem that includes all the major processor manufacturers, server companies, and operating system providers. These companies are working together to deliver enterprise-class virtualization functionality based on industry standards. Besides driving innovation

and building new solutions around the Xen standard, this ecosystem has also formed an extended testing team, further driving quality improvements.

Open Source technologies like Xen have a history of providing improved functionality, better performance, and lower total cost of ownership than proprietary technologies. Since Xen is free it's rapidly making its way into commercial offerings and end-user solutions. And as virtualization solution costs come down, it becomes feasible to deploy virtualization to every server throughout an enterprise IT infrastructure. History shows Open Source offerings, when generally accepted, tend to catch up with their proprietary counterparts quickly. Not since the Linux and Apache Open Source projects has such a large Open Source community and ecosystem formed so quickly. Although the current proprietary offerings have a few years' head start on Xen, the gap is expected to close quickly. The project and ecosystem has reached critical mass and the Xen hypervisor is emerging as the de facto standard. The tidal wave of innovation has begun.

Understanding Native Virtualization

Also relatively new to the market is what's known as, "native virtualization," which is a method that improves previous implementations by maximizing the benefits of the other approaches without the performance and management challenges. Previously, when deciding on a virtualization implementation, companies chose between operating system (OS) virtualization, full virtualization, and paravirtualization.

Native Virtualization is similar to full virtualization in how it supports a partitioned server running disparate guest operating systems "as is." This includes support for 32- and 64-bit applications and operating systems running concurrently. It also preserves investment in current certified software stacks, eliminating having to change or upgrade operating systems to run on the latest hardware. Although native virtualization is similar to full virtualization, there are major differences that improve efficiency and manageability. Unlike full virtualization, native virtualization doesn't rely on binary translation to emulate non-virtualizable x86 instructions. Instead it uses hardware virtualization assistance on the latest processors from Intel (Intel-VT) and AMD (AMD-V) to permit each guest operating system to run at full processor speed. Native virtualization

Virtualization is being overwhelmingly accepted in the market, **and this has caused the number of virtualization vendors to rise**

also doesn't require a complete instance of a host operating system to be installed and maintained. Instead, it uses small stand-alone virtualization services software running in a service partition to communicate with the hypervisor. Removing a complete host operating system greatly simplifies maintenance and management since there's no host operating system or virtualization software to install and maintain.

It's a common misperception that hardware-assisted virtualization minimizes the role and value of virtualization software. It's actually just the opposite. The new processors from Intel and AMD add other capabilities that greatly simplify and improve virtualization software performance. Without virtualization software, such as the Xen hypervisor and other virtualization services and virtualization management capabilities, you have only a standard server that can run one operating system.

Native virtualization leverages these hardware-assisted virtualization extensions to support virtualization software in an integrated and seamless fashion improving the efficiency, performance, and security of virtual servers. By providing a new privilege layer for virtual servers, and supporting key virtualization functions in hardware, this technology will simplify virtual server development and maintenance, improve interoperability with legacy operating systems, enhance security and reliability, and reduce the cost and risk of implementation. These extensions to the chip architectures will help commercial vendors deliver products that reduce the cost and risk of implementing server virtualization solutions and increase the reliability, availability, and security of applications running in virtual partitions.

Previously, companies chose from three differing proprietary approaches when deciding on virtualization for x86-based processors. One of the approaches is known as "full virtualization." Here the hypervisor provides a fully emulated x86-based virtual server where unmodified operating systems can run. Another implementation is "OS virtualization" in which a host operating system (single kernel image) multiplexes one operating system kernel to look like multiple operating system instances. A third approach is "paravirtualization" (partial virtualization), which uses slightly modified/customized versions of the operating system kernel to replace non-virtualized x86 instructions with virtualization APIs. All three of these proprietary approaches have advantages and disadvantages as they pertain to performance, efficiency, management, and maintainability. Challenges include lack of standards, performance overhead and degradation, and the need to modify operating systems in some cases. There's also complex management and the excessive administration overhead from maintaining virtualization software.

Choosing the Right Path

With the emergence of new virtualization technologies, the challenge for users is making sense of what is available in the virtual world and creating an environment that will deliver the promise of improved performance, reliability and total cost of ownership, while preserving investments in their existing software stack.

The benefits of standards-based products are well known and well understood. Customers benefit from "vendor choice," which reduces upfront and ongoing capital expenditures. With standards in place, IT managers can also tap a large pool of available professionals with required skill sets (e.g., Linux, J2EE, etc.). This reduces personnel costs and improves productivity. Other benefits include increased agility, flexibility, and interoperability. Industry standard solutions promote common approaches and architectures for business applications, making it easier to integrate new applications and functionality into core business processes and architectures. This interoperability promotes application agility and allows for a rapid response to changing business conditions.

It's difficult and rare to find an off-the-shelf product that delivers a total solution or precisely matches the features and requirements a business needs. It's usually necessary to integrate different software products and system management tools from different vendors. And integration is made easier by standard interfaces and protocols. A standards-based infrastructure also leads to a more stable environment because industry standards are typically backed by an ecosystem of vendors who support the standard and evolve it conservatively so as to not cause major disruption. The standardized environment increases the reliability of an infrastructure and reduces the time to repair it because support staff has fewer products to master and start from well known and documented capabilities.

Time To Virtualize

Virtualization is not only helping control costs and deliver the agility, manageability, and utilization that IT leaders covet, but it's also becoming a necessity in enterprises to control and maintain everyday activities. The single most popular use and often the initial application of server virtualization software is partitioning, which lets administrators put multiple virtual servers, each with its own unique operating system instance, on a single physical server. By doing this, IT administrators can consolidate their physical infrastructure, preserve their investment in existing operating systems and applications, and get more from their hardware investments. At the same time, more mature users of virtualization are getting additional business value from applying it as part of their provisioning, business continuity, and capacity management strategies.

Virtualization is being overwhelmingly accepted in the market, and this has caused the number of virtualization vendors to rise. Although it may become a daunting task to weigh the benefits of the list of vendors and options, IT organizations should simply look for solutions that leverage the advances and new technologies to further improve the ROI of virtualization. The rest will fall into place. ⦿

HP PROLIANT BL35p BLADE SERVER

with ProLiant Essentials Management Software

- Up to 2 Dual-Core AMD Opteron™ 200 Series processors
- High density: Up to 96 servers per rack
- Flexible/Open: Integrates with existing infrastructure
- HP Systems Insight Manager™: Web-based networked management through a single console
- Rapid Deployment Pack: For ease of deployment and ongoing provisioning and reprovisioning
- Integrated Cisco or Nortel switch options

HP STORAGEWORKS MSA1500cs

with StorageWorks Essentials Management Software

- Up to 24TB of capacity (96 250GB SATA drives)
- Up to 16TB of capacity (56 300GB SCSI drives)
- Ability to mix SCSI and Serial ATA enclosures for greater flexibility
- 2GB/1GB Fibre connections to host

Chaos, now under your control.

HP BladeSystem servers offer tools to help you keep pace with fluctuating demands. The HP ProLiant BL35p Blade Server is designed to relieve some of the stress. Its AMD Opteron™ processors offer dual-processor power with breakthrough efficiency. With management features like the Rapid Deployment Pack that lets you deploy and redeploy blades without missing a beat, and a single-view, graphical user interface that streamlines monitoring and configuration, HP BladeSystem servers work with you so you don't have to work so hard. And, bundled with the StorageWorks MSA1500cs, you can reduce the cost and complexity of deploying a storage area network, giving you a better return on investment.

SMART ADVICE > SMART TECHNOLOGY > SMART SERVICES

Call 1-888-220-7138
Click hp.com/go/bladesmag48
Visit your local reseller

OpenVista at Midland Memorial Hospital

Pushing back the barriers created by proprietary software

by Frank Pecaitis

This year, Midland Memorial Hospital in Midland, Texas, became the first community hospital in the country to adopt Open Source-based electronic health records (EHR). The implementation reflects the emergence of Open Source alternatives in healthcare applications as well as the growing movement to computerize patient medical records to reduce costs and improve patient care.

About the Author

Frank Pecaitis is vice-president of sales and marketing at Medsphere Systems, the leading commercial provider of Open Source technology to the healthcare industry. The company's OpenVista electronic health record system is a commercial implementation of the Open Source VistA electronic health record developed and used by the U.S. Department of Veterans Affairs.

frank.pecaitis@medspheresystems.com

At Midland, these two trends culminated in the deployment of Medsphere Systems' OpenVista, a Linux-based EHR platform with roots in the highly acclaimed Open Source VistA system developed by the U.S. Department of Veterans Affairs.

The software was installed on clustered HP servers running Red Hat Linux and was phased in over a seven-month period at two hospital campuses in this city of 95,000, located midway between El Paso and Fort Worth. The campuses are linked by a dedicated high-speed Gigaman circuit from AT&T enabling 24x7 access to a given patient's entire medical record by authorized clinicians from either facility.

The total cost of the installation was less than half that of systems from proprietary vendors, saving Midland millions of dollars.

"Linux and Open Source do not yet play a big role in hospital data centers, but OpenVista offered everything we wanted in an EHR system, we liked the fact that it could run on Linux-based HP servers because we're historically an HP shop, and we liked Medsphere's use of Red Hat Linux because that allowed most components of our technology stack to be Open Source," said David Whiles, Midland Memorial's director of information systems.

"Since we made the OpenVista decision, we have also decided to consider other systems that are certified to run on Linux," Whiles added. "We're looking for the best application regardless of the operating platform, but if the product we select offers a choice between Linux and a proprietary system, I will certainly choose the Open Source solution."

Catalyst of Change

Midland's move to Open Source and OpenVista began three years ago when the 320-bed not-for-profit hospital got word that support for its legacy patient accounting, registration, laboratory, and general financial systems would no longer be available as of 2006.

In the process of deciding how to replace those applications, Whiles and the rest of the hospital's information systems staff saw an opportunity to transition from paper medical files to a computer-based record. They started looking for a solution that would integrate all

aspects of patient care — from physicians' notes to prescriptions, X-rays, laboratory reports, and beyond — into a single electronic health record or EHR.

This EHR strategy has been endorsed by advocacy groups and even the White House because of a growing body of evidence indicating that computer-based medical records help improve the quality of healthcare while also reducing care delivery costs.

Studies show, for example, that submitting prescriptions electronically minimizes errors that stem from illegible handwriting. Other advantages range from a reduction in the interval between prescription writing and first medication to fewer adverse drug interactions, faster turnaround between test orders and test taking, fewer duplicate tests, and shorter hospital stays.

Traditionally, however, one of the major barriers to EHR adoption has been the enormous cost of buying and installing proprietary software. Open Source-based solutions like Medsphere OpenVista may prove to be the answer, particularly for the critical mass of hospitals and clinics that are financially challenged.

Open Source Economics

OpenVista is based on the VA's VistA (short for Veterans Health Information Systems and Technology Architecture), an electronic medical record-keeping system used at all VA facilities and credited with helping turn the VA into a national leader in quality patient care.

Leveraging the VA's multibillion-dollar investment in VistA and the fact that taxpayer funding put the source code in the public domain, Medsphere obtained the code under the Freedom of Information Act, ported the VistA software to Linux, removed VA-related components, updated the GUI, and made numerous functional enhancements.

The company has also commercialized the application and added professional services, ongoing product enhancements, and 24x7 technical support to provide the safety of a professional Open Source delivery model.

The resulting OpenVista EHR platform has the VA's 20 years of development and implementation at more than 1,300 sites behind it, providing a mature solution with real-world success, but

without the $18 million price tag that Midland encountered when investigating proprietary products. The total cost of the OpenVista implementation at Midland: $7.1 million.

That includes the platform's fully integrated suite of clinical and administrative modules, covering functions such as patient registration, medical records management, laboratory, pharmacy, radiology, mental health, nutrition, and food service. The Medsphere software also includes a clinical information system enabling physicians and other providers to document every patient contact, order tests, and proactively remind patients when they are due for follow-up exams or procedures.

"Without the economics of OpenVista," Whiles noted, "we most likely would not have been able to afford an electronic health record system at all."

Gearing Up

The implementation phase of the project began in January 2005 when Medsphere engineers embarked on customization and development work.

Medsphere's professional services team built roughly 30 interfaces to Midland legacy applications, including the hospital's Imagecast picture archiving and radiology information systems from IDX Systems Corporation (now part of GE Healthcare), the Clinivision respiratory services system from Puritan Bennett, and the Precision 2000 financial system from McKesson Corporation.

In June 2005, HP Professional Services began designing the hardware configuration, drawing on previous work with Medsphere OpenVista at a seven-facility long-term care organization in Oklahoma as well as years of providing support services for the VA's VistA system.

Installed two months later, the hardware infrastructure consists of a two-server Red Hat Linux-based HP ProLiant DL580 G2 cluster that runs the OpenVista system, a HP ProLiant DL380 server that functions as a quorum server for cluster administration, and an HP StorageWorks Enterprise Virtual Array 3000 for storing OpenVista data.

The clustered servers are equipped with HP ServiceGuard to provide the failover capabilities necessary to fulfill the hospital's stringent fault-tolerant requirements, and all servers are equipped with HP's Integrated Lights-Out Remote Management to enable troubleshooting from any location.

Midland also installed HP's OpenView Storage Operations Manager to manage the storage area network and an HP StorageWorks MSL5030 Tape Library for automated backup of all OpenVista data. Users access the OpenVista system from HP Compaq desktop
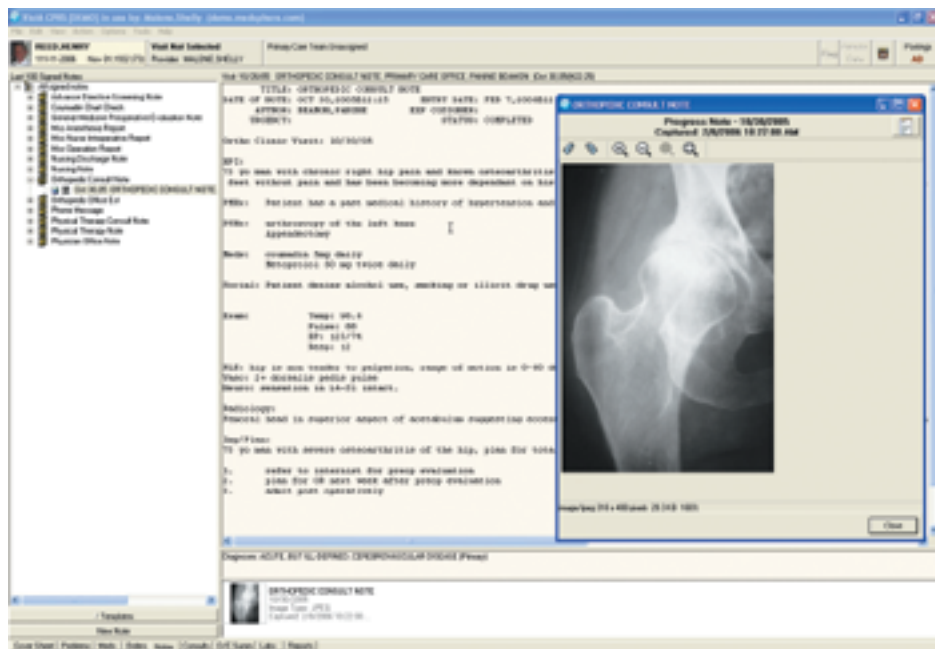
The name on this screenshot is fictitious to comply with government privacy regulations.

PCs equipped with Medsphere's Computerized Patient Record System client software.

Going Live

The OpenVista rollout to users began in October 2005 with the in-patient medications component of the system's Pharmacy module, designed to provide a comprehensive record of the medications used during a given patient's hospital stay. Although Midland's legacy pharmacy application was not being "sunsetted," the IT team elected to replace it to gain maximum integration benefits from the OpenVista platform.

A month later, Midland went live with OpenVista Laboratory, a module that lets users access information on all the laboratory tests done anywhere on the two-hospital campus, respond to alerts, and order additional studies from the OpenVista interface. This module replaced a sunsetted application.

The rollout then proceeded with the OpenVista Clinical Information System, enabling order entry by all hospital departments including lab, pharmacy, radiology, respiratory, physical therapy, and dietary personnel.

Activation of the hospital's clinical units began in March 2006 with the same-day surgery unit, followed by other units every one to two weeks. This added functions such as nursing documentation, physician documentation, clinical alerts and reminders, physician order entry, and scanning of clinical-related paper documents received from outside healthcare providers.

This phased-in deployment was accompanied by a rolling training schedule that trained users in each department one to two weeks before their go-live date. The entire hospital was

live on the bulk of the system by August 2006.

More work remains to be done. The roadmap includes building interfaces between OpenVista and other legacy applications, including Midland's surgery department management system from Per-Se Technologies and a perinatal monitoring system from GE Healthcare, as well as providing a means of remotely accessing the OpenVista database by physicians and other authorized providers who may not be on-site at the hospital. The hospital also will be deploying an OpenVista Bar Coding and Medication Administration module that helps reduce medication errors.

But Midland is already beginning to reap the benefits. Early on, for example, the exchange of information between the laboratory and pharmacy applications began proving useful for monitoring the effectiveness of drug administration. Users also began seeing faster access to lab results.

With a 2005 RAND Corporation report predicting that broad computerization of medical records will save $81 billion annually as well as improve patient care, and similar endorsements from other quarters, the healthcare community is actively looking for strategies to break its traditional dependence on paper charts. Midland's decision to embrace Open Source technology may pave the way for accelerating the migration to an electronic health record, in part by overcoming the cost hurdles of proprietary systems.

For that reason, this implementation will be closely watched in medical circles and potentially put Open Source on the medical map. It often takes only one believer to start a movement. Midland may play that role for open source EHRs. ⬤

An improvement

over distributing

flat files or relying

on RPC services
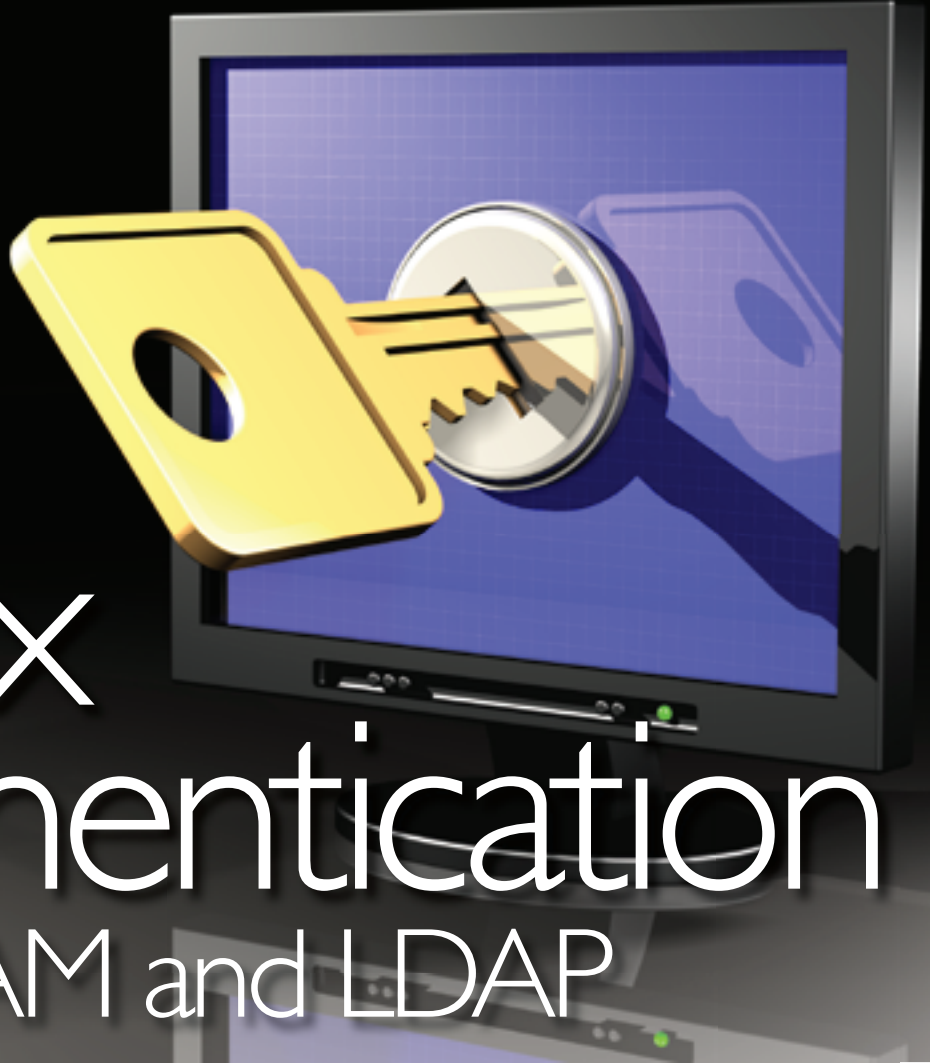
Linux
Authentication
Using PAM and LDAP

by Quan Dinh and Clayton Donley

I n an environment with more than a few Linux servers, managing users, groups, and other information securely across those systems is critical. Pluggable Authentication Modules (PAM) and the Lightweight Directory Access Protocol (LDAP) give administrators a way to accomplish this without having to distribute flat files or rely on RPC services, such as the insecure Network Information Service (NIS).

LDAP is a very mature protocol supported by everyone from Microsoft to IBM and Oracle. And reference implementations, such as OpenLDAP, have been available since the mid-'90s. PAM bridges the gap between traditional Unix and Linux login functionality and LDAP-enabled directory servers without requiring that applications or services be recompiled or reconfigured.

Aside from the obvious benefit of centralized storage and secure access, using LDAP has other key benefits depending on the server technology being used. These benefits generally include more granular password policy support, distributed or delegated administration of different groups of users in the same directory, and generally better performance and scalability than the alternatives.

Getting the Pieces

Most Linux distributions today support PAM out-of-the-box. However, the PAM_LDAP module, which is responsible for connecting the PAM framework to the LDAP-enabled directory service, usually needs to be downloaded and compiled separately. The source can be downloaded at www.padl.com.

PAM_LDAP itself has several dependencies, including SSL, LDAP, and SASL libraries, and it can be finicky about the specific versions used. While it's possible to install without some dependencies, such as SSL, not including them can render the system less secure. The most common issues with deployment are often related to getting a secure version of this module operational.

The directory that will be used to store and access account data is the other major component that will have to be installed. There are many options here and the choice of product is usually made on factors such as price, scalability, performance, and other key characteristics. Besides the base service, most commercial offerings include or have compatible tools to do everything from synchronize with other data sources to graphical user management and user self-service (e.g., password resets). Replicating data between servers and data centers is another key functionality that exists in most products, but in different ways.

LDAP-enabled directories have extensible schema that defines the type of data that can be stored. A standard set of schema, defined in IETF RFC 2307, may have to be imported or created in the directory to support the storage of Linux account information. In addition to accounts, this standard also defines schema for the storage of hosts, netgroups, services, and other types of data that has traditionally been stored in flat files, NIS, and related technology.

Making It Work

Once the modules are installed and the required directory servers are properly configured, the key steps remaining will be to migrate existing account information into the directory server and enable the PAM service.

Fortunately the first step is straightforward, given the ability of free migration tools from the same source as the PAM_LDAP module itself. These migration tools are capable of migrating all major information that would commonly be found in NIS or the equivalent flat files. The output of these files is a format called LDIF that can be used by standard LDAP tools for importing and modifying data in the directory.

With the data in place, the PAM module can be turned on and configured to point to the appropriate directory server. At this stage it's probably a good idea to keep a backup shell to help ensure that a mistake doesn't lock all the users out of the system.

Once authenticated, many applications will still need access to other data normally stored in passwd, group, and other files. This can be solved by configuring the system's name service switch configuration file (nsswitch.conf) to use LDAP. This defines the source and order that data ranging from account information to groups and even host name information will be checked in sources such as NIS, LDAP, and flat files.

Increasing Performance with Caching

Since many servers may be using the same directory, using a local server cache to reduce search and authentication requests may increase both performance and the number of servers that can be used with a single directory. This is especially true when using SSL and other overhead that can put an additional load on the directory servers.

In smaller environments this is less of a concern given that a typical directory server supports thousands of requests per second on modest hardware, but medium and large environments should look at using the Name Service Cache Daemon (NSCD). This service is included in most Linux distributions and requires no serious configuration. That said, it's important to realize that making a change to the directory, such as disabling an account, won't take immediate effect with the cache enabled.

Stronger Password Hashes & Security over the Wire

Many servers still support the CRYPT password hashing algorithm. This algorithm is easy to crack and a stronger algorithm, such as SMD5 or SSHA, should be selected instead. Access controls on the directory should still be configured to limit access to the hashed password.

Enabling either TLS/SSL or using SASL authentication mechanisms is an absolute requirement, since the alternative is to have the PAM module communicate with the LDAP directory server in a way that sends passwords in clear text over the network. Making this work is easier on some Linux distributions than others, but is an absolute must and can be accomplished on any system with the right combination of libraries as indicated in the PAM_LDAP and dependency documentation.

Conclusion

Together, LDAP and PAM provide a secure and scalable way to authenticate Linux users across multiple servers. Sites still using insecure and less scalable technology, such as NIS, can make the move to LDAP easily using existing modules and migration tools. Care needs to be taken to ensure that performance is maintained and that the security benefits of using the directory aren't sacrificed to bad practices such as weak hashing choices and transmitting passwords in the clear. ⦿

Useful Links
- *PAM LDAP Configuration*
  Unix Manual Page for PAM LDAP Configuration
  http://www.scit.wlv.ac.uk/cgi-bin/mansec?5+pam_ldap
- *LDAP entry*
  Directory Entries Administration
  http://ftp.unex.es/oradoc/application_server_10g/manage.904/b12118/entries.htm
- *NIS Schema*
  http://docs.sun.com/source/816-6682-10/oc_hpuxt.htm
- *http://www.oracle.com/pls/wocprod/docs/page/ocom/technology/products/oid/unix_pam_oid_wp.pdf*

About the Authors

Quan Dinh is a principal member of the technical staff at Oracle and specializes in network and Internet security. Dinh is currently responsible for the overall security of Oracle's Internet Directory and Virtual Directory products and has acted as a security advisor to Oracle's Data Center. He was previously responsible for helping to secure Oracle Application Server, Oracle Database, and Oracle SQL Net with PKI, SSL/TLS, and related security technologies. Dinh's other areas of expertise include Kerberos, SASL, and single sign-on technology. He was graduated from the University of California at Berkeley with a degree in computer science and mathematics.
quan.dinh@oracle.com

Clayton Donley is senior director of development at Oracle and currently responsible for Oracle's directory services product lines. Clayton's technical background in this area includes experience in IT, consulting, and development in addition to authoring the book LDAP Programming. Prior to Oracle, Clayton founded Octet String, Inc., a developer of virtual directory technology acquired by Oracle in 2005. Previous positions have included a wide range of roles at Motorola and IBM. Clayton has a BA from DePaul University in Chicago.
clayton.donley@oracle.com

Balancing Open Source and Commerce

Keeping the community committed and satisfying commercial customers

by Fabrizio Capobianco

I f ever there was a topic that someone was qualified to discuss, it would be me talking about how open source companies need to balance the interests of their community while making money. In fact, our company is named Funambol because it is based on the Latin words funis (rope) and ambulare (walking) that mean a tightrope walker. Managing an open source company requires constantly walking a tightrope that balances the needs of the community and the business. Every step involves decisions between keeping the community committed and satisfying commercial customers.

Choosing the right business model for an open source company is paramount for success. Even the largest and most well-known open source companies such as Red Hat and MySQL are still a work-in-progress. Yes, they have millions of users, and sales in the many millions of dollars. But they face constant threats to their business. What's to prevent another large entity from deciding that since Red Hat's code is open source, it can provide the same services as Red Hat? What's the guarantee that as MySQL gains even more free users that this will translate into profits? While there is no doubt that open source is a disruptive force to commercial software companies, open source companies themselves are susceptible to being "disrupted" as others can just take advantage of their openness.

Since we started Funambol five years ago, I have thought a lot about a business model that allows open source to be a viable alternative to proprietary software. As our website says, our "idea is to get the best of open source (high quality software, a community of people working together, no vendor lock-in), while providing a source of income to pay for the development of the software (yep, open source developers need to eat, too)." So what business model enables an open source company to achieve this balance?

To start with, I don't believe that the business model behind the Linux operating system is readily replicable. You cannot just build an open source project, hope it will gain a huge critical mass and expect a white knight to magically appear and contribute money. For an open source company to thrive and succeed, it needs a sustainable model that creates value for its community while generating enough cash to pay the bills.

Dual Licensing: A Duel Between Constituencies?

Open source companies can choose from many different business models and have many different ways to generate income, including charging for:
- Technical support
- Software functionality over and above what's in the open source project
- Training, consulting, professional (implementation) and other services
- Licenses for embedded software (such as DBMSs)
- Media distribution and documentation
- Hardware and appliances
- Legal indemnification, trademark licensing and other types of legal protection

The two predominant commercial open source business models are selling support and a dual license approach of offering a free open source and a commercial version of one's software. A major balancing act with dual licensing consists of where to draw the line between free and commercial functionality. A guiding principle for our company has always been to "just be honest" with our community. It is based on the concept that if someone changes the code in the project and uses the

About the Author

Fabrizio Capobianco, a serial entrepreneur and veteran executive at Reuters and Tibco, founded the first Italian Web company, Internet Graffiti. He also founded Stigma Online, developer of an information portal product with customers that included Kraft, Novartis, Italian Broadcasting Television, and the Italian Stock Exchange. Fabrizio has a PhD in computer science from the University of Pavia.

capos@funambol.com

project with the "public," they need to give the code back to the community or pay so that the money can be plowed back into the project.

Putting this theory into practice, however, requires constant decisions about major new features. Investors and salespeople want to "close source" important features to make it easier to sell the commercial version, while the community wants everything to be open. This creates tension in the community ("why did you put THAT feature in the commercial version? We need it! We'll build it ourselves").

Further, if you put too much in the open source version, customers will ask why they should pay for software; if you don't put enough in the open source version, you risk alienating the community as they might decide to work on more innovative projects or fork the project.

So, dual licensing with commercial extensions naturally creates tension between the open source and the commercial audiences that manifests in the product roadmap. In the long run, it's not clear that having commercial extensions is a sustainable strategy for many companies (what happens when the software reaches a state of maturity where it is good enough for most users' needs?). Some open source companies have acknowledged this and have switched to a 100% services-based model.

Segmenting Based on Users

So how can an organization remain faithful to its community's interests without creating a gigantic bait and switch situation, while generating enough cash to sustain itself?

In our case, we segmented our product based on our user base rather than product features. On one side, we work with our community to build a phenomenal open source project, pure and honest. On the other side, we build a commercial product based on the same core BUT targeted to the needs of someone else. If that "someone else" is not in the open source community, you are golden. Your community remains satisfied and gives back code to the project while "someone else" pays for it and gets what it needs.

Now the issue is finding that "someone else". In the Funambol case, the community is virtually everybody in the world, including individuals, universities AND enterprises. "Someone else" is mobile telecommunication carriers. The carriers have needs that are distinct from all other users. They target millions of users. They target consumers. It is the same core software but with different features that the open source community does not care much about. Segmentation based on user base makes it easier to plan features that eliminate the natural tension between open source and commercial versions.

Another open source business model consideration involves technical support. As time goes by and a category of software matures, there may be less of an opportunity to monetize it by selling licenses for additional functionality. Conventional wisdom in this case may be that a company can succeed by providing support and related services. But if you really have the best interests of the community in mind, your natural tendency should be to provide transparency and as much information about the project as possible, including documentation, troubleshooting tips, best practices, bug reports and more. The more resources the community has at their disposal, the more valuable the project becomes, leading to a self-reinforcing viral adoption of the software.

This reminds me of a story about a colleague who worked at one of the largest software companies in the world. He was responsible for planning the next version of their second largest product that generated hundreds of millions of dollars in software licenses and professional services per year. When he first started this project, he surveyed the product's users to find out what they wanted. At the top of their list was more information about how to use the product to accomplish complex tasks.

Upon reviewing this with the management team, they decided not to provide that information because they believed that it was this "hidden" knowledge that allowed them to generate major revenue. If the users could just read about how to implement the software by
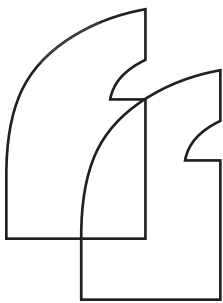
> for certain projects, it is a badge of honor to decipher complex code and **be one of the few people who understand how code works**

themselves, it would significantly reduce the money that the company could make. It was only after the customers bought the software that they realized their need for this type of information.

This practice of "information hiding" is subtly woven into the business models of commercial software vendors but it is not advocated for open source companies. The community could rebel if a company is not meeting their needs for technical information.

Related to this is a business model dilemma faced by open source companies vis-à-vis technical support and their mailing lists and community forums. On one hand, they want their communities to thrive and to actively share information. And it's important that the company's employees participate in this process, otherwise, the community could take on a life of its own and decide that they do not need the company. On the other hand, if the company spends too much of its resources to answer technical questions to please the community, why would anyone buy support? There would be no need to buy support as users can just get all of their questions answered for free. Once you start following a particular policy of what types of questions to answer for free, the community becomes conditioned to it and if you later decide to change course, you risk alienating the people whose loyalty and trust you are striving to keep. Most open source companies ultimately strike a delicate balance by providing tools such as knowledgebases and online documentation that enable users to help themselves, while charging for situations where users request individual attention.

Being Honest

Another fundamental business model question for open source companies involves the license for their software. Although there are a few predominant licenses such as Berkeley Software Distribution (BSD), GNU General Public License (GPL) and the Mozilla Public License (MPL), there are hundreds of other licenses and derivatives of these core licenses. Related to this, there is virtually no

case law to support the legality of the terms in many of these licenses (sort of like the wild, wild west).

The choice of which license to use for an open source project and company is of utmost consideration for the company's business model. Choosing which license is best for your project is beyond the scope of this article. However, there are many good resources that discuss the pros and cons of different licenses. It is also advised to consult with an experienced attorney who specializes in open source licenses.

That said, I'd like to share our own recent experience, as an example of licensing considerations that open source companies face and how it affects their business model.

Until recently, Funambol used the GPL. This enabled people the use of the software pretty freely, i.e., if they were just using it for internal use, they did not need to return anything to the community. If they distributed the software to the public, they were required to make their entire software available to the community or buy a commercial license that relaxed this provision. This is all in keeping with the community spirit of enhancing the code for everyone's benefit.

A situation we ran into recently involved the trend of delivering software as a service. In brief, some people were exploiting a loophole in the GPL relating to the term "distribution." Because they were distributing their service over the Internet instead of via a CD, they thought that they were exempt from the requirement to give code changes back to the community. We recently introduced a new type of license called the Honest Public License (HPL) that is equivalent to GPL but that says that software delivered as a service over the Internet is just another form of software distribution and therefore the company has to either make its code changes available to the community or pay for a license. This is a change that many other open source companies have applauded and is currently part of the latest draft of GPLv3. Our hope is that the final version of GPLv3 retains this language so that HPL will be unnecessary.

This is just an example of how the choice of license can greatly impact your business model so that you need to choose your licensing carefully.

Customer Balancing Act: Quality and Usability Are Key

Lastly, being an open source company that appeals initially to developers has other ramifications for a company's business model. Consider the quality and documentation of your software. It's one thing if the target user for software is developers. They may be forgiving in the face of errors, since they can fix code, or not mind about sparse documentation, because they can read the code and figure things out (in fact, for certain projects, it is a badge of honor to decipher complex code and be one of the few people who understand how code works). Now consider that you want to sell this same software to an audience of enterprise IT managers and corporate developers. These are people used to working with highly polished, documented and tested commercial software. They are less likely to be sympathetic with software that is buggy or appears unfinished. And it is pretty likely that open source developers may not care that much about polishing the software for corporate users.

Hence, a major tightrope challenge is how much effort you put into polishing the software for a commercial audience. As with the other decisions faced by open source companies, there are not universally right or wrong answers, you need to weigh the pros and cons for each situation. In some cases, it might take longer or require additional resources because of the need to satisfy multiple constituencies, which may run counter to the belief that open source software is faster and cheaper.

That said, there is little doubt that an open source company that makes it business model decisions with its community interests in mind all the time stands a much greater chance of being embraced by its community, which in turn will often benefit commercial customers alike. The trick is keeping the right balance on the tightrope. ⭕

Using Kernel Modules To Help Test Applications

Memory fragmentation & memory pressure

by Philip Copeland

Surely with its army of kernel hackers and proofreaders, the Linux kernel should be close to perfect. The reality is that most kernel hackers focus on only a piece of the kernel – VM, disk I/O, a particular driver, or some part of the network stack – and mistakes are made through oversight or because a certain condition wasn't given its due.

Everyone also has slightly different hardware and support chipsets, so it's not feasible for kernel writers to test on the entire range of available hardware. As a result, kernel writers depend heavily on accurate bug reports sent back upstream after mini-releases.

Because Linux is distributed as highly configurable source code, it needs to be tested with many different options to find combinations that stimulate bugs. Many operating system vendors only certify their product for a limited set of base options and hardware setups.

On the other hand, Linux supports a number of different instruction set architectures, different CPU variations for a given architecture, even mainframe processors (e.g., the S/390). Generally, developers don't own or have the resources to test on all of these different machine types. Furthermore, Linux supports a very large number of hardware devices, and there may be conflicts between different devices that can only be uncovered by widespread use by the public. As an interface between user applications and hardware, the kernel also needs to be tested by running multiple user-mode programs on it. This enables the testing of many combinations of system calls and other loads on the system.

Failure of a kernel in a production system usually has a greater impact than the failure of a user-mode program. If the kernel is flaky, machines can crash, the file system can get corrupted, and users can lose data and the use of their machine until they either reboot or the problem is resolved.

Worse, a buggy kernel can cause incorrect functioning of an otherwise reliable program. This kind of bug is insidious and can be very frustrating to track down.

There are also several things not directly related to the kernel than can go wrong, the most common being mismatched modutils for module loading. Testing kernels is almost always a significant undertaking; however, the reward is a better kernel. (Note: You should report bugs to the linux-kernel mailing list at linux-kernel@vger.kernel.org.)

Given a kernel, you want to test for the kinds of bugs that you should be on the lookout for in your code. You'll need to be on guard against heap corruption, buffer overflows, race conditions, failure to protect a critical memory region, and missing '=', '+' or '-' in comparison tests.

The most straightforward and methodical way to test this is by writing test tools that try out all of the different system calls, vary their parameters over the acceptable ranges, and ensure that the results returned are also within the documented range. You should also try making system calls with illegal parameters to ensure that an appropriate error code is returned.

There are some test tools available for Linux and the most significant is probably the Linux test project suite of programs at http://ltp.sourceforge.net/, which incorporates a comprehensive list of tools commonly used for testing the various Linux components. Most are concerned with getting expected results to known datasets or speed benchmarking of various routines. Nevertheless, these are important tests; however, you should be more focused on writing your own unique code for which no test has been written yet.

The main difficulty with testing for a reported bug is setting up an environment under which the bug can be exposed. Many tests try to artificially generate abnormal situations like heavy system load or heavy memory pressure, which don't represent a realistic situation. Quite a few bugs are found when people start using more extreme hardware configurations, such as 4GB+ of RAM, multiple network interfaces (three or four 4-port Ethernet cards), multiple Fiber Channel cards in a multi-initiator configuration, etc. These tend to be setups that aren't generally tested because they're not commonly found in a developer's environment.

Writing Tests

Let's assume a bug in your kernel code has been reported, which we determine must be

About the Author

Philip Copeland, a senior software developer in Oracle's Linux Engineering Group, has been working with Open Source software for more than 10 years.

philip.copeland@oracle.com

due to a heavy system load under memory pressure (i.e., there are several big processes running). We also suspect that somewhere in the kernel or in our code an assumption about memory allocation completing without checking the return value, or something similar, has occurred. It's something that hasn't shown up in any normal testing even though there have been three obscure reports about it. Needless to say, there's concern that something potentially harmful is happening.

Test 1: Memory Management System

Problems can arise from artifacts in the kernel memory management system. One well-known situation is memory fragmentation. Linux kernel memory management attempts to allocate memory requests in continuous groups, but where it can't, the memory ends up being fragmented. Occasionally, if an application makes a request for another chunk of memory, it may not be able to allocate it because there isn't room for a continuous memory allocation.

```
xxxx____xxxx____xxxxxxxx_____xxxxx____xxxx
```
(This is an overly simplistic model of the reality)

In the representation above, there are 3x4 page gaps, and one 8-page gap, but if 2x8 page-sized requests come in, only one will succeed despite there being 12 pages of memory left. This may leave the developer scratching her head wondering why the system is reporting sufficient memory but the application is experiencing memory starvation/out-of-memory problems. It's possible to cat ???????? /proc/buddyinfo to see how fragmented the currently free pages are. For example, a system that's been up and running for a while may look like this:

```
Node  0,  zone  Normal  2892 3014 65 23 1  0  0
1   1   1   0
```
(steps up in 1,2,4,8,16,...)

On this system, 2,892 single pages of memory could be allocated immediately, 3,014 pairs exist, but only 65 groups of four pages can be found. If something comes along that needs a lot of higher-order allocations, the available memory will be exhausted quickly, and those allocations may start to fail even though in real terms there's sufficient memory.

That describes what might be happening, but from a debugging point-of-view how can this situation be mimicked? One way is to just play out a normal run of whatever the system does and hope the error happens again. Unfortunately, this could take weeks. Instead, something to artificially generate the fragmentation is needed. This isn't easily done from a userspace program either, so it will be necessary to turn to some kernel module programming to generate the situation.

http://oss.oracle.com/projects/codefragments/src/trunk/fragment-slab/

When run, this module can artificially starve the system of contiguous pairing and provides a means to test an application or kernel code in a memory-hostile environment.

Test 2: Memory Pressure

Another persistent problem is a situation known as memory pressure, which often leads to a system slowing to a dreadful crawl or, worse, locking up under load. One of the fundamental sources for memory pressure is the file-system page-cache usage, along with the buffer_head entries that control them.

Another problem area are inode and dentry cache entries in the slab cache. Linux struggles to keep both of these under control. User space processes provide another obvious source of memory use. These are partially handled by the OOM killer subsystem. This is all further complicated by the swap process, which, in an effort to increase available memory, is used by the system to kick out items considered to be 'idle' processes or process memory. One of the easiest ways of generating this situation is

to simply reboot the box using the mem= boot time option to reduce the amount of memory available. Since one of the main sources is file-system page-cache use, a kernel module can be written to artificially generate a load through allocations via the kernel buffer_head_allocation/deallocation routines.

http://oss.oracle.com/projects/codefragments/src/trunk/bufferheads
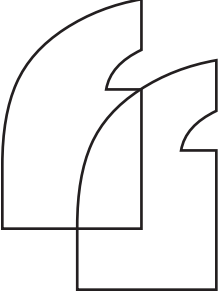
It's possible to artificially increase or decrease the number of buffers used by plugging in values on the command line when the module is loaded or by modifying the values via the /proc interface. These modules aren't complex. The most complex thing about them is the ability to dynamically change values on-the-fly without having to unload/reload the modules.

The main advantage of using a kernel module to create the test environment for memory fragmentation is that we can create a fairly realistic environment in a few seconds as opposed to having to wait days or weeks for the same situation to occur naturally. Trying to recreate the same situation using a normal user land ????? program would be thwarted by the kernel's memory management system trying to stack all the requests in a manner that defragments the system on-the-fly.

In the second example, the main advantage is that we can do in a few seconds what would take quite a while in user land ????? (Have you ever tried to create a million files on an FS before? It takes a VERY long time). So while it's not impossible to create the environments from user land ????? programs, the kernel module option offers a means to create extreme conditions quickly and at a low cost.

These are only two examples relating to memory use. This could easily be extended into creating artificially hostile network environments or a crippled IO system. You simply need to know in advance what type of conditions you want to create for your application to run in. ●

The most significant set of Linux test tools is probably the Linux test project suite of programs, but you should really be writing your own unique code for which no test has been written yet

Open Source Software, Standards, and Java

Sun Microsystems publishing Java under the Open Source license

by Stephen R. Walli

Sun Microsystems recently announced its intentions of finally publishing Java under an Open Source license. But what does that actually mean? We'll take a quick look at what it means to be ''Open Source,'' how the Java language specification compares to other more formal language standards, and the importance of the brand and certification programs. We'll then look at what benefits Sun may get from distributing Java as Open Source and at some of the problems that will have to be addressed.

About the Author

Stephen Walli is vice-president of Open Source development strategy for Optaros (www.optaros.com), responsible for architecting and managing Optaros' relationships with the Open Source community. Before joining Optaros, he was an Open Source advocate at Microsoft, where he focused on "shared source" business strategies and was responsible for the technical implementation of Open Source-related community projects such as creating a business model at Microsoft to engage in the Open Source community. Stephen was a business development manager on the Windows Platform team, working in the space between community development, standards, and IP concerns. Prior to Microsoft he was vice-president of R&D and co-founder of Softway Systems, Inc., a venture-backed start-up that developed the Interix environment to re-host Unix applications on NT. Stephen has also worked as an independent consultant for X/Open, SunSoft, Unisys, and the Canadian government. He was once a development manager at Mortice Kern Systems and a systems analyst at Electronic Data Systems. He was a long-time participant and officer of the IEEE and ISO POSIX standards groups, representing both USENIX and EurOpen (EUUG) and has been a regular speaker and writer on open systems standards since 1991.

Open Source Software

The Open Source Initiative defines Open Source software and a license must meet 10 criteria to be considered an "Open Source software" license. Essentially, it's a way of thinking about licensing software. It boils down to some very simple ideas about access to source code and the ability to modify the software and distribute those modifications. It encompasses the concept of Free Software as defined by the Free Software Foundation around a set of software "freedoms," such as the freedom to study how a program works and adapt it to your own needs.

Open Source software is typically developed in a collaborative community, either under a strong leader who coordinates the development community, or a meritocratic process where a developer earns the leadership role in the community like the process favored by the Apache community. Some companies build businesses based on Open Source software projects, generally ones they control. For example, MySQL (the company) maintains MySQL (the database engine).

In these cases, the software has a copyright, is owned, and is therefore licensable. Free and Open Source software is not "public domain" in any sense of the phrase.

Software developed in successful Open Source collaborative communities shows all the hallmarks of well-developed software from other processes. Essentially good software is developed by good software developers regardless of the licensing strategy. So Open Source software has just as much potential to be well-structured, have well-defined stable interfaces, and be delivered through a disciplined process that encompasses software inspection, mandatory version control, and automated building and testing as software developed in other ways.

Where Open Source differs from other well-developed software is in the collaborative community. The best developers interested in the software can participate in its creation and evolution regardless of where they live or work. This provides a number of benefits:
- Many people see the source code. Software inspections regardless of how informal prove to be much more effective at finding bugs than testing.
- The code is used and tested in a broad base. This expansion of the "test" bed tends to shake out bugs faster and hardens the software.
- A diversity of expertise and experience can be leveraged. This applies both to improving the code base, as well as to innovating on the code base to take it in new and interesting directions.

Sun has been an active participant in the Open Source software world since its inception. The original SunOS operating system was a Berkeley Unix derivative that came out

of the collaborative efforts around Unix in the early days. Sun has contributed heavily to the accessibility features in the Gnome desktop. Most recently it has opened the Solaris source code base under its own OSI-approved Open Source license (the Common Development and Distribution License or CDDL) and has been developing the OpenSolaris community. So Sun definitely has experience with Open Source, both contributing to other and developing its own communities.

Let's shift gears for a moment and take a look at Java and the Java Community Process from a standards perspective, as that has been Java's history to date.

Standards (Open and Otherwise)

A specification is simply a document describing some interface for interoperability. Lots of companies publish specifications to enable customers and partners to interoperate software with their products better. In such cases, where the specification is published by a single commercial entity, it typically benefits the company by encouraging add-ons to its product.

A standard is a specification that has been put through some form of consensus process by a collection of interested parties. It may be a formal government-supported de jure process with checks and balances to ensure that the consensus isn't anticompetitive collusion. It may be an industry or trade organization (CBEMA, ECMA, IEEE) with a broad interest in an area, e.g., computing standards. It may be an industry group with a narrower focus (e.g.. OASIS, W3C, IETF). The consensus process has rules that define such things as participation, acceptance, interpretation, amendment, and withdrawal.

The economic purpose of a standard is to encourage and enable multiple implementations of the subject specified, i.e., it is the opposite of a company specification that directly benefits the company's own ecosystem. A standard is designed to increase choice and benefit consumers. A successful standard has to have multiple implementations that conform and interoperate. If there's only one implementation then it's just a vendor specification, regardless of the process it was put through to get a stamp of approval.

Sun created the Java Community Process (JCP) to manage and maintain the evolution of the Java language. While it's easy to claim that the JCP is "controlled" by Sun, the JCP actually has more in common with an industry group building standards than a vendor-controlled specification to enhance a vendor's own ecosystem. The JCP has members well beyond Sun. It has a well-defined, consensus-based process to manage the myriad Java-related specifications. Membership is open to all to participate. Its purpose is to encourage multiple implementations of Java and not simply add-ons to Sun's Java world.

The hard part of any standards organization is how best to measure and warrant that an implementation conforms to the specifications to protect the value of the standard's brand in the marketplace. How does one best signal to the marketplace that a subject is what it claims to be? Conformance measurement and certification is an expensive process. Sun, through the JCP, has put in place an expensive process to certify that Java technologies delivered by anyone do indeed meet the specifications.

Certifications are always taken on by the group that stands to gain the most (or lose the most in some cases). Essentially the group that cares makes the investment and develops a program to certify things against the standard.

POSIX was a standards effort defined by the IEEE that undertook to define an operating system interface in the C language to sup-
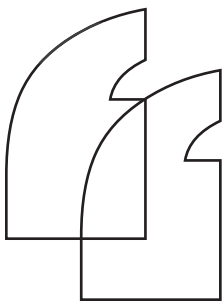
> Open Source Java hasn't been driven by Sun, but demanded by the community around Java; whether the demands are valid or not, or indeed politically motivated or not, **there are still good things to be had from the process**

port application portability. The IEEE didn't handle conformance measurement however. The U.S. National Institute of Standards and Technology developed the certification to support U.S. government procurements. The IETF skips expensive certification processes after the fact, and instead uses the reality that they define networking standards to require that an RFC has two independent interoperable implementations to gain standards status. The people who economically need measurable conformance take responsibility for putting the system in place.

So too is the case with Java. This is as true for proprietary product specifications and their certification programs as it is for industry and de jure standards. The vendor stands to lose the most with respect to its proprietary specification's brand in the ecosystem, as any industry standard has to gain from the value of demonstrating that multiple implementations conform.

Standards AND Open Source Software

Standards exist to enable multiple implementations of a technology. Open Source software to a certain extent represents the one true implementation of a technology. When there's one true implementation there's no need for a standard. For example, there'll never be a Perl language standard. But the interaction is actually more subtle.

Standards typically occur in mature spaces where there's a wealth of experience and expertise. When it comes time to create a technology standard, the vendors in the space will pick a shared technology base from which to create a standard. Every vendor would love to claim its technology is the standard, and often make claims to having the "de facto" standard, essentially such a ubiquitous technology that it's a "standard in fact." The real world doesn't work that way however, regardless of how it's marketed. When true standards are delivered, they come from a shared technology base so that none of the

participants feels another has a market-dominant starting position.

The standard will differ from that core shared technology base, but not so much that the shared base doesn't quickly morph to conform to the new standard, and the collection of vendors can quickly bring products to market. In today's world, Open Source software projects represent that shared technology base out of which standards can be delivered to facilitate multiple implementations.

This is a somewhat odd position then for Sun with Java. As the keeper of a primary reference implementation, and the creator of the standards development organization, it would seem Sun is in an odd place. Indeed, it's almost as if the process is working backwards.

Open Source Java

So what will Open Source Java mean? First remember that this effort hasn't been driven by Sun, but demanded by the community around Java. Whether the demands are valid or not, or indeed politically motivated or not, there are still good things to be had from the process.

Sun has driven the Java standardization process through the JCP for some time and has a strong collaborative community and process, supported by a strong certification and branding program. Delivering Java technologies as Open Source still makes sense, however, even if the standard has led the implementation so to speak.

As a primary reference implementation, it will provide the following benefits to the entire Java community, Sun included:
• It will harden the primary implementation for Sun's and the community's benefit. Allowing others to tinker and explore will uncover new and interesting problems, which can then be addressed.
• It will enable new innovation. Many claim Java's day is done. Allowing new implementors to explore the primary production base will invariably lead to new ideas and

innovation on the platform.
• As new code enters the source base, it will likely come in at a very high level of quality. Even if Sun developers act as the primary committers for the foreseeable future a high level of inspection will be brought to bear on code coming in from the outside. For new work delivered from the inside, the inspection by the community will likely be equally vigorous.

That is not to say that there won't be challenges. As with any large code base that exists in a commercial product, all will have to be inspected carefully from a number of angles. Obviously Sun won't want any immature code released, but Sun also has to ensure that all code licensed in from the outside can be released and manage that process.

Sun already supports a strong development community around OpenSolaris and hopefully that experience can be leveraged by the "OpenJava" team. Likewise, Sun already supports a strong collaborative community in the Java Community Process, so it has a great channel to begin its Open Source efforts when it figures out how it intends to publish what sources. It began the release of Java EE 5 with the GlassFish project, and now time will tell if it can harness all its collective experience in Open Source software, standards, and the JCP to bring about a complete Open Source Java world. ⊙

References
• *The Open Source Initiative and the Open Source definition can be found at* http://www.opensource.org/docs/definition.php.
• *Free software as defined by the Free Software Foundation can be found at* http://www.fsf.org/licensing/essays/free-sw.html.
• *Sun's CDDL license can be found at* http://www.opensource.org/licenses/cddl1.php.
• *The Shared Source CLI can be found at* http://msdn.microsoft.com/net/sscli/.
• *The Mono project can be found at* http://www.mono-project.com/.

Hello Dali!

An introduction to the Eclipse Dali Java Persistence API tools project

by Shaun Smith

On June 26, 2006 the Eclipse Foundation announced the availability of new releases of 10 Open Source projects. This simultaneous release event, named Callisto, garnered a lot of attention for the 10 projects involved. But, meanwhile, on the same day and without much fanfare, not even a press release, the Dali JPA Tools project shipped its first formal release numbered 0.5. With the release of Dali 0.5, developers now have a solid set of tools for developing applications for the new Java Persistence API (JPA) in Eclipse.

The Java Persistence API

The Java Persistence API is part of the new Java EE 5 EJB 3.0 specification and defines a vendor-neutral standard for object-relational mapping. But don't be fooled by the term "EJB." The JPA specification was certainly developed under the umbrella EJB 3.0 specification, but that doesn't mean it's just for Java EE. JPA is designed to work in Java SE as well as EE, and will likely be split off into its own specification in the future.

JPA defines a way to map plain old Java objects (POJOs), not Entity Beans, to relational databases. This means you can use JPA to store the Java objects you write without having to subclass a JPA-provided class or implement any JPA interfaces. One of the driving goals of the JPA specification was ease of use and it shows.

JPA in Eclipse

One of the most striking features of JPA is the use of Java 5 annotations to define object-relational mappings. By adding annotations to your classes you can make instances persistent. JPA uses the term "Entity" for persistent objects and uses the @Entity annotation to identify them. This means that you can use a simple text editor or the Eclipse Java editor to work with JPA.

Unfortunately the Java editor doesn't understand what the annotations mean. As far as it's concerned annotations are just metadata markup. It can validate the syntax but not the semantics. For example, in Figure 2 the Phone Entity's number field is mapped to a column

Figure 1: When using Java 5, the Eclipse Java editor is aware of annotations and will perform code completion on annotation and property names

Figure 2: The @Column overrides the default mapping of the 'number' field to the 'NUMBER' column

named "NUM." That column may or may not exist in the database but without JPA-aware validation you won't find out until runtime — a very bad time to find out. This is essentially what Dali provides: JPA-aware tooling and validation to ensure that what developers build at design time will run at deployment time.

Dali Overview

Dali provides tools to develop JPA ap-

About the Author

Shaun Smith is co-lead of the Dali JPA Tools project and a principal product manager at Oracle for TopLink, the basis for the open source TopLink Essentials JPA reference implementation.

plications targeted at either Java SE or Java EE and supports top-down, bottom-up, and meet-in-the-middle development approaches. Regardless of whether you want to persist an existing Java object model, manipulate data in an existing database, or connect your existing Java classes with an existing database, Dali can improve your productivity and help ensure that you don't spend your time in an endless edit, deploy, run, debug cycle.

For example, Figure 3 shows the same Phone Entity as Figure 2. But when using Dali, a problem is found in the JPA mapping for the number field. Dali has validated the column name specified in the @Column annotation against the Phone table and found that there's no such column.

JPA Defaults

One of the most useful features of the JPA is its defaulting rules. For example, if an Entity is not explicitly mapped to a table then the table name defaults to that of the Entity. Defaulting rules let developers "program by exception." That is, they only need to add annotations for things that don't match the defaults. In the case of our Phone example, Dali has confirmed that a table exists in the database with the name "Phone" — the same name as the Entity. Since there's no problem, no errors are displayed.

Dali Views

Dali contributes two views to the Eclipse user interface along with a perspective that defines a layout suitable for performing object-relational mapping. Those two views are the Persistence Outline and Persistence Properties.

Persistence Outline

The Persistence Outline view is similar to the Eclipse Java Outline but offers a JPA view of your object. In Figure 4 the Persistence Outline shows the Phone Entity and its mapped attributes. In JPA you can either put your mapping annotations on a Class's fields or properties (JavaBean style getters). The Persistence Outline displays the mappings the same way regardless of which of the two approaches you choose. Using the outline you can get a quick thumbnail sketch of the mappings for an Entity, even if those mappings are spread throughout the Java source file. For Phone you can see the id holds the primary key of the Entity and is a Basic mapping — mapped directly to the database column. The number attribute

is also a Basic mapping while the custs attribute is a collection of objects mapped as a ManyToMany.

By default, the Persistence Outline selection is linked with the Java editor so you can navigate quickly around a Class to individual mappings. The linking is reciprocal — selection of attributes in the Java editor will also update the selection in the Persistence Outline. This quick navigation to mappings is useful if you want to jump to them in the Java source editor, but is more useful when paired with the Persistence Properties view.

Persistence Properties

The Persistence Outline gives you a brief summary of your mappings and lets you navigate between them, but doesn't offer any help in editing your mappings. That's the function of the second view contributed by Dali: the Persistence Properties view. In Figure 3, we saw how Dali validated mappings and put error markers in the Java editor and errors into the problems view. But as the saying goes, acknowledging you have a problem is just the first step. The Persistence Properties provides tools for understanding and resolving mapping problems.

The Persistence Properties view performs a couple of very useful functions. It shows how a mapping is configured and, perhaps even more importantly, it shows the defaults that will be applied by a JPA runtime when the Entity is deployed.

For example, in Figure 5 the column mapping for the number attribute is defaulting to True for insertable and True for updatable. Defaulted values are clearly visible through the use of the word "Default." Notice that the column name isn't marked as a default value because the developer has explicitly specified it in an annotation.

But let's return to the problem Dali identified with the number attribute — there's no such column as NUM in the Phone table. A valid column name has to be selected, and the Persistence Properties view can help. It offers valid options for all mapping settings including settings that require access to the data model.

In Figure 6 the column name dropdown contains all the Phone table columns. It also displays what the default column name would be if nothing were specified. Since the default is correct, the default may as well be used. With the entire mapping using default values Dali removes the mapping annotation from the Java source to keep it uncluttered by unnecessary annotations.

Figure 3: Dali validates JPA mappings against the data model. In this case there's no column named NUM on the Phone table
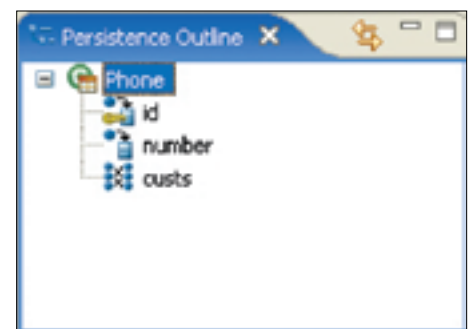

Figure 4: The Dali Persistence Outline View

Figure 5: The Persistence Properties View displays the settings for mappings explicitly defined by the developer and the defaults that are implicitly applied by JPA runtimes
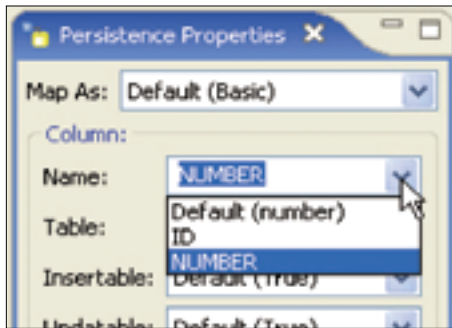
Figure 6: The Persistence Properties View provides access to the data model for mapping
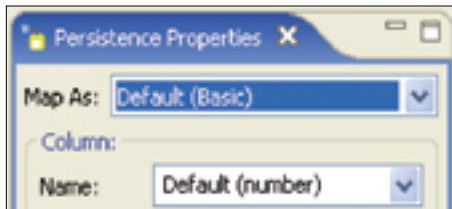

Figure 7: Figure 7 The Persistence Properties View showing the default Basic mapping of the number attribute and the default column name column

Figure 8: With no mapping annotation on the number attribute Dali validates the defaults as problem-free

Figure 7 and Figure 8 show that with no column specified, in fact no mapping specified at all, the defaults validate against the data model and there are no problems.

Top Down and Bottom Up

We've seen how Dali can help with the "meet-in-the-middle" approach of mapping an object model to an existing database but two other approaches are supported. Using Dali, it's possible to start with a set of Java classes annotated as Entities and generate the database tables they map to. Generation of Entities from tables is also supported.
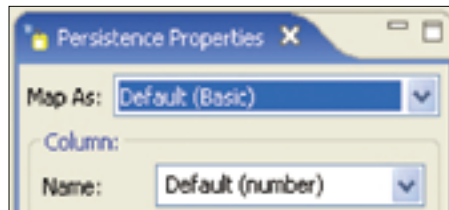

Figure 9: Basic persistence.xml created by Dali

Figure 10: The list of Entities can be updated in the persistence.xml using the 'Synchronize Classes' menu option
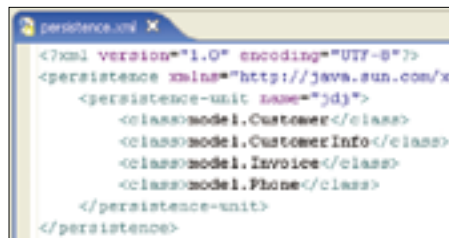

Figure 11: The persistence.xml file with a list of Entities updated by Dali

The algorithm Dali uses for both Entity and table generation is defined by the JPA default mapping rules with a few extra heuristics to deal with differences in Java/database naming styles like underscores versus camel case. Dali's support for generation offers a quick way to bootstrap a new JPA application. You can generate Entities or tables to get a starter configuration and then refactor either knowing that Dali will flag any breakage in your mappings with problem markers.

Deployment

Deploying a JPA application is straightforward whether you're using Java SE or EE. Dali doesn't offer any specific packaging and deployment support beyond some assistance with maintaining the persistence.xml file (more on this below), however, deploying JPA Entities is just like deploying POJO applications. You can jar them up using the standard Eclipse support for exporting jars or include them in an Enterprise Archive (EAR) as a utility jar using the Web Tools Platform (WTP).

Persistence.xml

The one XML configuration file required in the JPA specification is the persistence.xml file. This file defines important runtime settings including database connection information

and transaction type. When you add persistence support to a Java project, Dali creates a basic persistence.xml file and places in the src\META-INF folder. Typically you'll hand-edit this XML file to reflect your deployment configuration.

As mentioned, JPA applications can be deployed to both Java SE and EE environments. However, when running outside an EJB 3.0 container, JPA requires an additional piece of information in the persistence.xml: a list of all the persistent Entities. In the 0.5 release Dali provides support for keeping the persistence.xml in sync with your defined Entities.

Right-clicking on the persistence.xml file in the Package Explorer and selecting Java Persistence>Synchronize Classes will update its list of classes (Figure 11).

Future Directions

The focus of the Dali 0.5 release was annotation-based mapping and support for the core JPA mapping types. Dali 1.0 will offer editing and validation support for both annotation and XML-based mapping as well as the use of XML mappings to override annotations as defined in the JPA specification.

Smoother integration of the Dali tools with WTP is also a high priority for 1.0. The Dali project is now incubating inside WTP as one of the new Java EE 5 technologies that will be incorporated into WTP 2.0.

In 1.0, Dali will also leverage the enhanced database support provided by the Data Tools Project (DTP). The combination of WTP with Dali and the DTP will provide a comprehensive toolset for the development of Java applications that rely on relational data.

Getting Started

The best place to begin with Dali is to visit the project home page, check out the online demos, download the plug-ins, and go through the tutorial. The Dali newsgroup is monitored by the development team and is a great place to ask a question or get help.

And finally, like every Open Source Eclipse project, contributors are welcome. Contributors meet on the dali-dev@eclipse.org mailing list to discuss technical issues and make decisions. ⊙

Resources
- *Dali home page with links to downloads, documentation, project roadmap, and tutorials: http://www.eclipse.org/dali.*
- *TopLink Essentials JPA Reference Implementation: http://otn.oracle.com/jpa.*
- *JSR 220: Enterprise JavaBeans 3.0 specification: http://www.jcp.org/en/jsr/detail?id=220.*

Complex JAVA J2EE Hosting made easy.

WebAppCabaret sm

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

1.866.256.7973

JAVA J2EE-Ready Managed Dedicated Hosting Plans:

Xeon III
SAMEDAY SETUP

Dual 3.2 GHz Xeons
4GB RAM
Dual 73GB SCSI
1U Server
Firewall
Linux
Monitoring
NGASI Manager

$**399**
monthly

Pentium 4 I
SAMEDAY SETUP

FREE SETUP

2.4 GHz P4
2GB RAM
Dual 80GB ATA
1U Server
Firewall
Linux
Monitoring
NGASI Manager

$**199**
monthly

2nd month FREE

4Balance I

1 Database Server
and 2 Application
Servers connected
to 1
dedicated
load
balancing device.
Dual Xeons.
High-Availability.

$**1724**
monthly

NEW! Geronimo 1.0 Hosting . Virtual Private Servers(VPS) starting at $69/month

At **WebAppCabaret** we specialize in **JAVA J2EE Hosting**, featuring
managed dedicated servers preloaded with most open source JAVA technologies.
PRELOADED WITH:
JDK1.4 . JDK1.5 . Tomcat . Geronimo . JBoss . Struts . ANT . Spring . Hibernate
Apache . MySQL . PostgreSQL . Portals . CRM . CMS . Blogs . Frameworks
All easily manage via a web based control panel.
Details:
-All Servers installed with the latest Enterprise Linux
-Firewall Protection
-Up to 60 GB daily on site backup included at no extra charge per server.
-Database on site backup every 2 hours
-Daily off site database backup
-A spare server is always available in case one of the server goes down
-Intrusion detection.
-24x7 Server and application monitoring with automatic self healing
-The Latest Bug fixes and Security updates.
-Tier 1 Data Center. 100% Network Uptime Guarantee
-Guaranteed Reliability backed by industry-leading Service Level Agreements

Log on now at **http://www.webappcabaret.com/jdj.jsp** or call today at
1.866.256.7973

NGASI POWERED

WebAppCabaret sm

JAVA J2EE Hosting

Eclipse Foundation Approves Aperi Storage Management Project

(Ottawa) - The Eclipse Foundation, an open source community committed to the implementation of a universal development platform, has announced that it has approved the creation of the Aperi Storage Management Framework Project. The Aperi Project, which was proposed in June 2006, will give customers more choices for deploying open storage infrastructure software, based on an industry-standard framework developed by the open source community.
www.eclipse.org/aperi

AccuRev Announces SCM Usability Enhancements to Eclipse Plug-In

(Lexington, MA) - AccuRev Inc. has announced significant new features for its AccuBridge for Eclipseplug-in. AccuBridge for Eclipse provides developers utilizing their familiar Eclipse IDE integration platform a transparent, stable and highly functional environment to directly access AccuRev best-of-breed software configuration management (SCM) functionality to increase the agility, reliability and quality of their complex, geographically distributed and parallel development process.
www.accurev.com

Red Hat Champions a New Public Forum for Software Companies

(Raleigh, N.C) - Red Hat, a provider of open source solutions to the enterprise, has announced its continuing commitment to security services with a new initiative, implemented by the National Institute of Standards and Technology (NIST), that enables members of the software industry to officially and publicly comment on vulnerabilities. This service is being implemented within the National Vulnerability Database (NVD) at NIST, based on Red Hat's recommendation. As a widely recognized, comprehensive cyber-security resource containing all publicly available U.S. government vulnerability information, the NVD can be used by users of both open source and proprietary software. By centralizing and communicating information for vulnerabilities, customers and users will benefit from increased information coming from both the U.S. government and vendors.
www.redhat.com/security

Terracotta Offers Eclipse Plug-in for Point-and-Click Clustering Technology

(San Francisco) - Terracotta, Inc., a provider of solutions for enterprise Java scalability, has announced the availability of its Eclipse plug-in for Terracotta DSO, the company's enterprise-class JVM clustering technology. Bundled with Terracotta DSO, the new plug-in makes Terracotta's point-and-click clustering functionality available from within the Eclipse IDE and demonstrates the company's on-going commitment to open source integration.
www.terracottatech.com

Black Duck Software to Review Contributed Code

(Waltham, MA) - Black Duck Software, a provider of software compliance management solutions, has announced that the Eclipse Foundation, an open source community committed to the implementation of an extensible development platform, has purchased and deployed Black Duck's protexIP/development platform. Eclipse uses protexIP to review software submitted by committers and ensure it is in compliance with the specific software licensing requirements of the Eclipse Foundation.
www.blackducksoftware.com

Alfresco Extends Leadership with Production Ready Open Source Business Process Management

(London) - Alfresco Software Inc., a provider of an open source enterprise content management solution, has announced the availability of the Preview Release of Alfresco 1.4. This is a major release that expands both the depth and breadth of Alfresco in the $3.9BN Enterprise Content Management (ECM) market.
www.alfresco.com

IBM Unveils New Offerings for the Open Source Community

(Cambridge, MA) - Reinforcing the company's strategy around open source business models, IBM has announced new software and resources to help customers leverage their investments in open source development tools and resources. The new offerings will help software development teams build Eclipse-based open integrated development environments that are technologically diverse and vendor neutral.
www.ibm.com/software

2X Opens Source Code to Its NX Based Terminal Server for Linux

(Frankfurt, Germany) -2X has announced the release of 2X TerminalServer for Linux, an open source terminal server for Linux, which enables users to run a Linux desktop and Linux / Windows applications over any type of connection. The 2X TerminalServer is based on the open source NX X-Windows compression protocol. X-Windows is bandwidth-intensive and therefore does not scale well or run over low bandwidth connections. The NX protocol on the other hand, compresses the X-Window protocol and enables users to run a complete Linux desktop and Linux / Windows applications over slow dial-up links. On LANs, it means a much higher number of clients can be supported without affecting network speed.
www.2x.com

Centrify DirectControl Enables Automated Trading Desk to Increase Security and Simplify Linux Server Administration

(Mountain View, CA) - Centrify Corporation, a provider of solutions that securely integrate non-Microsoft systems and applications with Microsoft Active Directory, has announced that Automated Trading Desk, LLC (ATD) will use Centrify DirectControl to simplify its server administration and increase its security on a network of over 600 SUSE Linux Enterprise Servers from Novell. By centralizing the management of servers in Microsoft Active Directory, ATD will eliminate server-by-server identity management and access control, allowing it to enforce best practices across its entire environment.
www.centrify.com

Coraid Delivers Intelligent RAIDShield to Protect Storage

(San Clemente, CA) - Helping to improve disk integrity for Linux storage, Coraid has announced it has added RAIDShield disk protection to its ATA-over-Ethernet (AoE) storage appliances. RAIDShield technology improves the reliability of RAID volumes through an advanced disk fault detection and recovery algorithm, eliminating the risk of data failure on high density SATA disk storage systems.
www.coraid.com

The 451 Group Introduces the 451 Commercial Adoption of Open Source (CAOS) Research Service

(New York) - The 451 Group, a New York-based technology-industry analyst company focused on the business of enterprise IT innovation, is introducing a first-of-its-kind service that investigates both user experiences and vendor strategies as enterprise customers begin to deploy open source software. This new analytical service - the 451 Commercial Adoption of Open Source (CAOS) Research Service - was designed to help enterprise end users, software vendors and investors track and understand the opportunities and threats presented by open source.
www.the451group.com

Pointsec Unveils New Version of Encryption Software for Linux

(Chicago) - Pointsec, a provider of the de facto standard for enterprise security software for laptop and desktop PCs, PDAs, and smart phones, has announced the latest version of its endpoint encryption software for Linux desktops and laptops, Pointsec for Linux 2.0. Committed to its continued support of Linux-based operating systems, Pointsec now offers Linux users new features including support for remote help, single sign-on, and broader security for all system files starting from a pre-boot authentication. Pointsec has expanded its products to provide out-of-the-box endpoint encryption software for standard Linux kernels on desktops and laptops, reducing both cost and risk for organizations. The newest version of Pointsec for Linux continues to provide an endpoint encryption solution with an even lower total cost of ownership, through improved usability and management features that ease the burden on administrators. With its support for standard Linux kernels, users and administrators are not required to rebuild the kernel in order to work with Pointsec for Linux 2.0.
www.pointsec.com

Instantiations Broadens RCP Developer Support for Linux Platform

(Cambridge, MA) - Instantiations Inc., a provider of advanced software development solutions, has announced version 2.0 of RCP Developer, its development tool for creating Eclipse Rich Client Platform (RCP) applications. The latest version adds Help Composer and extends Linux support for RCP Packager and WindowTester. RCP Developer accelerates the creation of Rich Client Applications by harnessing the power of Eclipse RCP to bring Java back to the desktop. Announced last March, it is the first development tool that enables developers to exploit Eclipse RCP core services for creating Java-based client and standalone desktop applications.
www.instantiations.com

www.AjaxWorldExpo.com

AJAXWORLD™
CONFERENCE & EXPO

SANTA CLARA SILICON VALLEY

SYS-CON Events is proud to announce the first-ever AjaxWorld Conference & Expo 2006!

The world-beating Conference program will provide developers and IT managers alike with comprehensive information and insight into the biggest paradigm shift in website design, development, and deployment since the invention of the World Wide Web itself a decade ago.

The terms on everyone's lips this year include "AJAX," "Web 2.0" and "Rich Internet Applications." All of these themes play an integral role at AjaxWorld. So, anyone involved with business-critical web applications that recognize the importance of the user experience needs to attend this uniquely timely conference – especially the web designers and developers building those experiences, and those who manage them.

CALL FOR PAPERS NOW OPEN!

We are interested in receiving original speaking proposals for this event from i-Technology professionals. Speakers will be chosen from the co-existing worlds of both commercial software and open source. Delegates will be interested learn about a wide range of RIA topics that can help them achieve business value.

The Best of Both Worlds

Debugging Linux/Unix code using Windows XP software development tools

by Anatoly Krivitsky

These days the size of almost all the resources available to a programmer (memory volume, CPU speed, etc.) are on the rise except for one: the time required to complete a project, which is shrinking. So it's important to work with tools that are convenient and make you productive. It's also important to keep the cost of software development in check by keeping as much as possible of the investments that were already made.

Windows – and currently XP Home and Professional are the most widespread versions – has a huge number of installed copies that can be measured in the hundreds of millions. Linux, on the other hand, is supported by a dedicated community of developers and often provides high-quality software. At the moment Unix operating systems (like Solaris 9 and 10) are widespread for servers.

That's why questions related to inter-system communication between Windows and Linux/Unix (including but not limited to virtualization) are attracting growing attention from developers and businesses.

A good chunk of software development tools for XP is customizable so you can create a configuration (hot keys, macros, menus, commands, customizable syntax, highlighting, etc.) that suits your specific needs. Many developers invested time in such customization.

Suppose you work for a Windows shop and put a lot of time in customizing such tools and then your shop starts a Linux or Unix project. Well, this article will teach you how to save your investment and apply Windows XP software development tools to debugging Linux or Unix code with the originally developed technology of inter-system communications between Windows and Linux/Unix.

Making Your Own IDE

You're not going to compete with big companies like Microsoft, IBM, or Borland, just make yourself a convenient tool. However, manufacturers will be able to apply the proposed technology to their integrated development environments (IDEs) or other software development tools (say editors with extended syntax checking) as described below. Let's consider how your typical IDE works, gen-

erally speaking. The programmer creates/edits code using an IDE editor then complies it using a build-in or external compiler that's called with a menu selection (like, for example, the Build | Build solution) and gets compilation results right in front of his eyes in the IDE. Now, why not to apply the same idea to a mixed Linux or Unix-Windows XP development environment? This article will show you how to do it.

What If the External Compiler Works Under a Different Operating System

In our case the external compiler works under a different operating system. The system for the compiler can be run in the same computer as XP (if virtualization is used) or on a different computer on the other side of the planet. It really doesn't matter as long as you can transfer the code from your favorite XP software development tools to the external compiler that works under Linux/Unix and get back compilation results.

General Description of Proposed Technology

The idea is simple. We'll use the following components that together will compose a technology for transferring code from the XP software development tools to Linux/Unix, run the Linux/Unix compiler and get back the results of the compilation:
- A browser on the Windows XP side that will be automatically pointed to a Web page described below from the Windows XP software development tool. You will find a description on how to do it.
- A Web server that runs the different operating system (Linux or Unix). The operating system in question will be referred to as the server operating system. In fact, any operating system that can host the appropriate Web server can be used with the proposed technology. The Web server just has to be able to handle a FILE input type in an HTML form and some kind of server-side software (Perl scripting, PHP, servlets, JSP, etc.) that's capable of performing the operations described below. Most Web servers these days (like Apache and Sun One) easily fit this requirement.

About the Author
Anatoly Krivitsky is a PhD in computer science and has more than 25 years working experience in IT. He is the author of two dozen published papers and books and holds five patents. For more information see http://www.myjavaserver.com/~akrivitsky/.
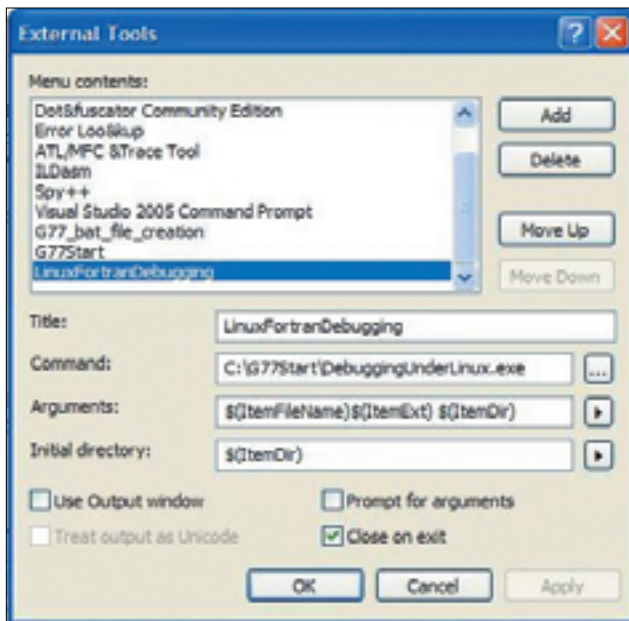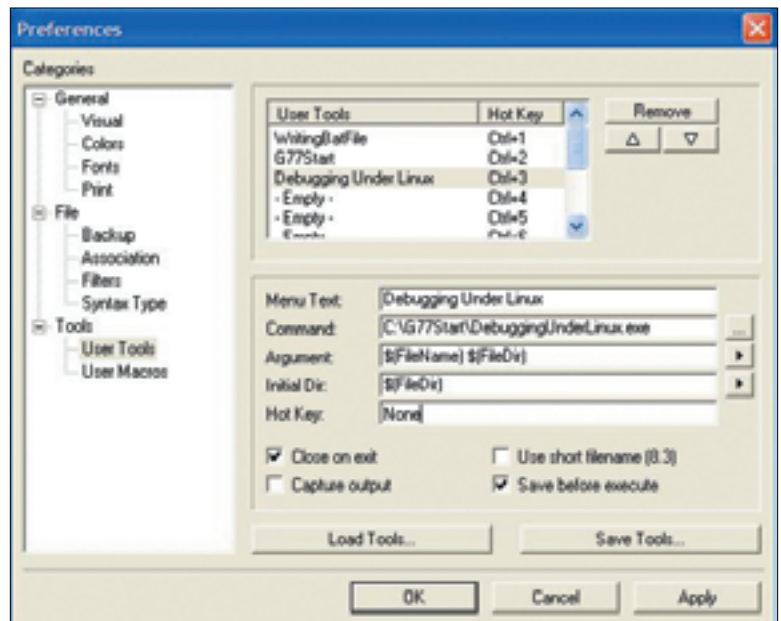*akrivitsky@yahoo.com*

Figure 1

Figure 2

- A specially designed Web page that will be automatically filled with code from the Windows XP software development tool that will be put in a FILE input type of the appropriate HTML form. This page will be filled out by a special application that is described in the next bullet. It will be deployed on the server operating system, but will be accessible from XP and serve as a "bridge" between the two.
- An application that is called from the Windows XP software development tool and automatically fills out the appropriate fields on the Web page. The application will use the Web browser of your choice to do that. In the example below the application is written in Visual Basic.NET (in a way that's compatible with VB.Net 2003 and 2005). But, of course, any language that lets you send keys to the browser (consider C#, for example) can be used for the application.
- Server-side software that gets the code from the Web page, writes it into a file on the server operating system, calls the external compiler in the system, and sends the results of the compilation back to the browser. In this article a Perl script will be used for this purpose as an example but, as mentioned, one can use any number of technologies (servlets, JSP, PHP) to do the same.

Enough theory, let's do something for real.

Putting It All Together: An Example of the Described Technology

Suppose you use Fedora Core 5 running under XP via VMWare Workstation 5 or 5.5 (i.e., virtualization will be used). Please see my LinuxWorld magazine article "Running Fedora Core 5 Under Windows XP – Tips and Tricks" on virtual machine settings and other useful information (http://linux.sys-con.com/read/219966.htm).

For your convenience comments on how to use this technology under other conditions are included in this section. The Web page that will be used is named RemoteDebugging.html. It's given in Listing 1. Note that only standard HTML tags constitute a page that's usable with most Web servers these days. If you use a remote server you'll need a special page (or special fields on the page) with a logon ID and password and/or other means of authentication that will be automatically filled out with the required credentials in a way similar to other fields that are used in a form. Please see the details below on how the form will be filled out. Security issues related to authentication are a huge topic that's far beyond the scope of this article. If you are new to security, a good place to start is to google the Internet and read the book on secure coding listed in the resources. You'll get many other security references from it.

The Web page with its needed permissions will be put in the appropriate directory of the Web server. If you use the Apache server that comes with Fedora Core 5 with the standard configuration, the directory will be /var/www/html. You may want to read your Web server documentation on where HTML pages should be put if you use other Web servers in Fedora Core 5 or if you use a remote Web server. In the case of a remote Web server you'll need server administration assistance if you're not allowed to ftp (or otherwise submit) the Web page in question to the server and give it appropriate permissions.

The Perl script that will do the job described above is called RemoteCompilation.pl. It's given in Listing 2. With the standard Apache in Fedora Core 5 it should be put under /var/www/cgi-bin. Note that in this case files that are produced by scripts from /var/www/cgi-bin have an owner named apache that belongs to a group with the same name. You may want to issue the command: chown [your user name goes here]:apache RemoteCompilation.pl after you create the script to be able to edit the script and run it. You must replace [your user name goes here] with your Fedora Core 5 user name. Then you'll have to issue the command: chmod 775 RemoteCompilation.pl (or set up permissions with another appropriate tool like Konqueror). Of course you should have enough authority to run these commands. In case of a remote server (or a server other than the Apache under Fedora Core 5) you'll have to read the documentation to figure out what the name of needed group is. If a remote server is used, you'll likely have to contact server administration to figure out what the appropriate permissions for the script are and how to set it up since security is involved here.

Note that with a remote server, the appropriate code for authentication checking must be used just after print start_html in the script (or implement separate software for this purpose). Only standard Perl constructions are used in the script so it's possible to use the script with almost any current Web server. First, the script checks if a standard output

of your compiler exists (in the example it's a file called a.out) and deletes it. The script also deletes an auxiliary file named err.txt that will be described below.

After that it reads the code for debugging from the appropriate parameter and the file name for the code from another parameter and writes the code to the file with this name to the server. Chunks of 512 bytes are used for writing. If host operating system and/or compilers require it, you should then insert a statement for changing the file permissions. The needed compiler will be called by the script using the file in question as well as another special auxiliary file with possible errors called err.txt to which any errors produced by the compiler should be redirected.

In our example Intel Fortran for Linux is used. But the same technology can be applied to any needed compiler. If you use other compilers you'll have to read the documentation and adjust the script accordingly (particularly the statement that calls the compiler). In our case, redirection is provided by 2>err.txt in the corresponding statement. In case of other compilers figure out from the documentation how to provide the redirection. Note that the script files are written in the directory in which the script is located. In the case of a remote server you have to contact server administration to figure out the appropriate directory. Then the script checks if a.out (the default name for the executable file with this compiler) is produced, which means that the compilation is OK. The check is done using a pipeline of ls and grep. If the compilation is OK, the user will see a corresponding message in the browser. If compilation failed (i.e., a.out isn't produced) the script will read each line from err.txt and display it in the browser. Note that in this case the HTML <br> tag will be added to the end of the line. With other compilers you may want to use another condition to determine whether or not compilation is OK. The condition can be found by reading the documentation.

The application that calls the browser with the page in question from your XP software development tool is given in Listing 3. The name

of the .exe file obtained with the source code in Listing 3 is DebuggingUnderLinux.exe. If you use VS.NET 2005 to compile the source, a System. Windows.Forms reference should be added to the "Console application" type of the project. The DebuggingUnderLinux.exe in our example is placed under C:\G77Start, but you can use any directory. Note that in the example the application calls FireFox from a specific location, but you can easily adjust it to any browser with any needed location. If you use FireFox, it should be closed before the application runs again.

The application is called from the tool by the menu, hot keys, etc. and obtains from the caller 2 parameters: the name of the file that contains the code and the path to the directory where the file is located. Most modern software development tools allow these two parameters to be passed via a customizable configuration (please see the examples below). Consider the following popular software development tools under Windows XP: VS.NET 2005 and Crimson from http://www.crimsoneditor.com. The example of the configuration to be used with the application from VS.NET 2005 is given in Figure 1 and the one from Crimson is given in Figure 2. As soon as configuration is made you can debug the needed code with these tools simply by clicking the hot keys of your choice (CTRL+3). If you use Crimson select Tools | Debugging under Linux. If you use VS.NET 2005 you can use any name for this command and assign hot keys to it. Note that with a remote server, the URL of the Web page used in the application will be given to you by server administration.

If you use VMware workstation 5 or 5.5 for virtualization and Fedora Core 5 as the guest operating system, the URL can be obtained by the following means. Within the Fedora Core 5 terminal window, issue an ifconfig command. Locate "inet addr" in eth0. It will be part of the url. In this example the part in question is 192.168.227.129 and this value is used in the application. You need to obtain the value in question using ifconfig and adjust the application accordingly.

Please refer to Listing 3 on how to get the URL using the part.

To test of the example of this technol-

ogy you can use dwhet.f (an application for performance measurement from my Web site at http://www.myjavaserver.com/~akrivitsky/ G77.html. (You may want to read my article in JDJ at http://jdj.sys-con.com/read/83099.htm on the subject.) First use it "as is." You will see a "Complilation OK" message in your browser. Then introduce an error by typing "error" anywhere in the code. You'll see a corresponding error message in the browser.

What We Just Did

This example shows you how by simple and therefore effective means (a Web page, a Perl script, and an application) you can debug Linux or Unix code with Windows XP software development tools. In other words, we can run the compiler under another operating system and get the results back to Windows XP with the proposed technology.

Homework for Manufactures

This example offers a solution that fits the needs of many developers. However, it can be polished further by IDE manufactures and included in their products. Here's how. After obtaining results in the browser one sends the following commands to the browser window: CTRL+A to select all the text in the window, CTRL+C to copy the selection to the clipboard, and ALT+F4 to close the window. See Listing 3 on how to send commands to the browser window. Then you can grab the text from the clipboard to your IDE and deal with it in exactly the same way as output from your native IDE compiler for the convenience of your user. The user will be able to get additional help on errors, and get to the line with the error by clicking the corresponding error message. So by using this approach you can give your IDE additional multi-operating system capabilities. ◉

Resources
- *M.G. Graff and K.R. van Wyk. Secure coding: Principles and Practices. O'Reilly and Associates. 2003. ISBN 0-596-00242-4.*
- *My Web site has useful information on the subject http://www.myjavaserver.com/ ~akrivitsky.*

Listing 1. RemoteDebugging.html

```
<html><head>
<!--
  Copyright (c) 2006 Anatoly S. Krivitsky, PhD.
  All rights reserved.
  Conditional permission for free use of the
  code.
  As soon as above copyright notes are
  mentioned,
  the author grants a permission to any person
  or organization
```

```
to use, distribute and publish this code for
free.
Questions and comments may be directed to the
author at
akrivitsky@yahoo.com and akrivitsky@gmail.com.
Disclaimer of Liability
The user assumes all responsibility
and risk for the use of this code "as is".
There is no  warranty of any kind associated
with the code.
Under no circumstances,
including negligence,
```

Visit the *New*

www.SYS-CON.com

Website Today!

The World's Leading *i*-Technology News and Information Source

24/7

FREE NEWSLETTERS
Stay ahead of the i-Technology curve with
E-mail updates on what's happening in your industry

SYS-CON.TV
Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!
Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST
Streaming video on today's i-Technology news, events, and webinars

EDUCATION
The world's leading online i-Technology university

RESEARCH
i-Technology data "and" analysis for business decision-makers

MAGAZINES
View the current issue and past archives of your favorite i-Technology journal

INTERNATIONAL SITES
Get all the news and information happening in other countries worldwide

JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

IT Solutions Guide

Information Storage+Security Journal

JDJ

Web Services Journal

.NET Developer's Journal

LinuxWorld Magazine

Linux Business News

Eclipse Developer's Journal

MX Developer's Journal

ColdFusion Developer's Journal

XML Journal

Wireless Business & Technology

Symbian Developer's Journal

WebSphere Journal

WLDJ

PowerBuilder Developer's Journal

```
<title>Remote compilation of the code</title>
</head>
<body>

<h2>Remote compilation of the code</h2>
<form method="post" action="/cgi-bin/RemoteCompilation.pl"
       enctype="multipart/form-data">
       Code for uploading will go here: <p>
       <input type="file" name="code" size=40><br>
       A file name to be sent will go here: <br>
       <input type="text" name="filename"><br>
       <input type="submit" value="If this button is clicked, the code will be
remotely compiled">
       </form>

</body>
</html>
```

Listing 2. RemoteCompilation.pl

```perl
#!/usr/bin/perl –w

# Please see listing 1 for copyright notes and disclaimer of liability
use CGI qw(:standard);
use Fcntl qw(:flock);
use strict;

print header;
print start_html("Remote compilation");
print h2("Remote compilation");
if(-e "a.out") {
system("rm a.out");
}
system("rm err.txt");
my $file_name = param('filename');
my $code = param('code');
   print "Code: $code<br>\n";
   open( CODE_ON_THE_SERVER, ">\./$file_name" )
     or &dispaly_error(" The $file_name in current directory was not opened for
the following reason: $!");
   flock( CODE_ON_THE_SERVER, LOCK_EX );
   while ( read( $code, my $chunk_512, 512 ) ) {
       print CODE_ON_THE_SERVER $chunk_512;
   }
   close(CODE_ON_THE_SERVER);
   print "$file_name has been created on the server from $code <p>\n";

print "<H2>Remote Compilation Results </h2>";
# 2 is used for Linux std.err redirection
system("ifort -O $file_name 2>err.txt");
```

```perl
my $compilation_ok = "no";
my @lines = `ls -l | grep "a.out"`;
foreach my $line (@lines) {
$compilation_ok = "yes";
print "Compilation Ok <br>";
}
if ($compilation_ok == "no") {
open(ERRORFILE, "err.txt");
while(my $record = <ERRORFILE>) {
print $record."<br>";
}
close(ERRORFILE);
}


print end_html;

sub dispaly_error {
    my ($error_message) = @_;
    print "<h2>The following error occurs</h2>\n";
    print "$error_message<p>\n";
    exit;
}
```

Listing 3. Source code of the application

```vbnet
' Please see listing 1 for copyright notes and disclaimer of liability
Imports System.Windows.Forms
Imports System.Threading

Module Module1

    Sub Main()
        Dim arguments As [String]() = Environment.GetCommandLineArgs()
        Dim FileName As String = arguments(1)
        Dim pathToDirectory As String = arguments(2)
        ' delete possible \ at the end of pathToDirectory
        If (pathToDirectory.EndsWith("\")) Then
            pathToDirectory = pathToDirectory.Substring(0, pathToDirectory.
Length - 1)
        End If
        Console.WriteLine("File Name = " & FileName)
        Console.WriteLine("pathToDirectory = " & pathToDirectory)
        Dim FileNameAndPath As String = pathToDirectory & "\" & FileName
        Dim FireForID As Integer
        FireForID = Shell("C:\Program Files\Mozilla Firefox\firefox.exe
http://192.168.227.129/RemoteDebugging.html", AppWinStyle.NormalNoFocus)
        Threading.Thread.Sleep(2000)
        AppActivate(FireForID)
        SendingKeys("{TAB}")
        SendingKeys(FileNameAndPath)
        SendingKeys("{TAB}")
        SendingKeys("{TAB}")
        SendingKeys(FileName)
        SendingKeys("{TAB}")
        SendingKeys("{ENTER}")
    End Sub
    Sub SendingKeys(ByVal s As String)
        Threading.Thread.Sleep(100)
        SendKeys.SendWait(s)
        Threading.Thread.Sleep(100)
    End Sub

End Module
```

accomplishments and share their knowledge (look at the blog phenomenon), and because the development of a piece of software that's useful and well written is something to be proud of, people usually share it. Then, when someone else sees the value of the pet project, he or she is likely to not only use it, but also share improvements that can benefit the original author and the additional contributor. Clearly, it's a more efficient practice than commercial models. Red Hat and Novell, for example, both fund development of the Linux kernel by paying the salaries of developers. They each focus on adding value in other areas as well, like testing and other quality assurance measures, efficient delivery of a commercial version operating system, and technical support. After all, there's little incremental value between the two base operating systems. It is a much more efficient model than say, commercial Unix, where each vendor duplicates efforts in areas that add very little incremental value.

In other industries - especially businesses that have similar needs and whose competitive advantages are based on assets unrelated to information technology infrastructure - shared development can help the whole industry. Unlike commercial software, open source software is driven from the bottom up, which means it's not typically developed by independent software vendors (ISVs) and then distributed. However, we are starting to see that happening, as well as more and more software vendors turning over their software to the open source community, just like database vendor Ingres has done. Some do it because they don't want to maintain the software, but the smart ISVs know that innovation comes from the end user. Mature products can receive new life from the open source vendor community and adapt to an ever-changing IT landscape.

Invariably, leading commercial software in a variety of software segments will be challenged by open source markets. The following are current trends I believe will outlive the fad moniker:

- **Virtualization:** It's what's hot right now, but there's no slowdown in sight. Over the past ten years, our thirst for faster computers drove the industry to provide blazingly fast chips that sipped more power and threw off a lot more heat. Now, those efforts are being penalized by data centers that can't meet electrical and cooling needs. Virtualization allows consolidation and higher utilization on Linux servers. Commercial software vendor VMware (an EMC company) was one of the fastest-growing software companies in the past five years. The Xen project, a hypervisor, which is a piece of software that allows multiple operating systems to run simultaneously on the same server, is the open source alternative to VMware. Open source virtualization is being advocated by more than just Linux users. Even Microsoft has announced they will be supporting Xen even though it might be competitive with their Microsoft Virtual Server.
- **Open source systems management:** Systems management is an established software sector used pervasively across many industries that have been paying dearly for expensive seat licenses and up-front royalties. New monitoring companies like Zenoss (www.zenoss.com), GroundWork Open Source (www.groundworkopensource.com), and Hyperic (www.hyperic.com) are offering subscription-based monitoring products that offer very similar products at attractive pricing models. Some of these projects use the same open source projects to build their commercial offerings. Because they are sharing the burden of development, they don't have to fund all the R&D on their own, and they can bring products to market quickly and inexpensively, which enables competition with the Big 4 management vendors (HP, CA, IBM, and BMC).
- **Open source business intelligence and planning software:** This

is more of a prediction than a trend, but it's one of the next sectors to be challenged by upstart open source projects that are replacing traditional enterprise software vendors. It's an industry ripe for open source: long-established vendors who have a broad customer base and a need for customization. Pentaho (www.pentaho.com), an open source business intelligence platform, will start to challenge companies like Business Objects, SAS, and Hyperion. In Enterprise Resource Planning (ERP), the open source company Compiere (www.compiere.com) is going to challenge industry leader, SAP. Both companies will offer a competitive price and an easily integrated solution that will be a viable option to the entrenched vendors.

If you are still skeptical about the longevity of commercial viable open source software, that's fine. There's no place to check the open source scorecard and, truthfully, you could and might conduct all your computing on proprietary closed source software. Of course, you will probably end up with greater expenses from software royalties and a lack of flexibility. As I write this article I am using a mixture of open source and commercial software, but as I look back 10 years ago, I didn't use a single free piece of software as part of my desktop computing regimen. I don't recall even using the term open source or reading about open source in the mainstream media, whereas now, lengthy conversations on the topic are the norm for technophiles and business people alike. Open source has transcended fad status and gone beyond being a trend. It is the evolution of a better way to develop and maintain software, and many of us would be well served to see how we could capitalize on it. ⊙

AJAX and the Maturation of Web Development

From "View Source" to "Open Source"

by John Eckman

I have always imagined the information space as something to which everyone has immediate and intuitive access, and not just to browse but to create."

(*Weaving the World Wide Web* by Tim Berners-Lee)

From the beginning, the World Wide Web that Tim Berners-Lee imagined was a place where the architecture of participation ruled. Berners-Lee's first application for accessing the information Web was both a browser and an editor, and throughout the early 1990s he worked diligently to encourage Web browser development groups to develop editors and servers as well as browsers. As early as the spring of 1992, the challenge was clear: "Although browsers were starting to spread, no one working on them tried to include writing and editing functions....As soon as developers got their client working as a browser and released it to the world, very few bothered to continue to develop it as an editor" (*Weaving the World Wide Web* by Tim Berners-Lee).

Developers tended to defer the editing functionality for a number of reasons, mostly to compress development schedules and to get the browser out the door - because they felt many people, if they didn't need it, would at least use it - without the editor, which was more complex and useful to a smaller audience. Netscape Communicator 4.0, released in 1997, did finally include Netscape Composer, although its licensing terms only allowed free use for non-commercial purposes. Internet Explorer never contained an editor directly, though Microsoft acquired FrontPage from Vermeer in 1996; FrontPage 1.0 had been released in 1995 (http://www.seoconsultants. com/frontpage/history/).

It wasn't just the complexity of the editing functions themselves, of course, but also the fact that reading pages required a much simpler authorization model, in which the user either has access to the document or does not. In fact, the cluster of issues at hand - from version control of Web pages to multiple authors editing the same page, sometimes at the same time, to control over who should have access to change what pages - would busy the content management industry for the better part of the next decade.

While the early Web browser teams deferred creation of an HTML editor, they retained a key element of Sir Berners-Lee's original Web browser/editor:

The 'View Source' menu item migrated from Tim Berners-Lee's original browser, to Mosaic, and then on to Netscape Navigator and even Microsoft's Internet Explorer. Though no one thinks of HTML as an open source technology, its openness was absolutely key to the explosive spread of the Web. Barriers to entry for "amateurs" were low, because anyone could look "over the shoulder" of anyone else producing a Web page ("The Architecture of Participation" by Tim O'Reilly).

This "View Source" menu item, which was not buried in developer editions or professional versions but was part of the core browser, created a culture of easy access to knowledge.

As the complexity of presentation-tier Web development grew, with Cascading Style Sheets, JavaScript, and DHTML in the mix, the View Source culture of Web development evolved into an open source culture of frameworks and libraries. It is this culture that enables the viability of current AJAX-based approaches to Web development.

The View Source Culture

From a narrow perspective, the decision to include a View Source option in the Web browser was an insignificant choice, perhaps useful for troubleshooting formatting issues, but of interest to a very small community. As Berners-Lee puts it, "I never intended HTML source code (the stuff with the angle brackets) to be seen by users. A browser/editor would let a user simply view or edit the language of a page of hypertext, as if he were using a word

About the Author

John Eckman leads the Next Generation Internet Practice at Optaros (www.optaros.com), and has over a decade of experience designing and building Web applications. He received a BA from Boston University and a PhD from the University of Washington at Seattle; he expects to complete an MS in Information Systems from Northeastern University in 2006. He also achieved PMP certification in 2003.

processor. The idea of asking people to write the angle brackets by hand was to me, and I assume to many, as unacceptable as asking one to prepare a Microsoft Word document by writing out its binary coded format" (Weaving the World Wide Web by Tim Berners-Lee). What View Source did (and still does!) was let users who were interested in learning to create Web pages see what HTML source was delivered to the browser to produce the page currently being rendered. Perhaps because many of the early Web users were developers of one kind or another, it became an expectation that any reasonable browser would include the ability to View Source.

Viewed more broadly, however, the View Source command was nothing short of revolutionary. It set the expectation that users should be able to not only view the "rendered" document, but also the "code" that created it. Because early browsers often differed in their interpretation of HTML, this was critical. Significantly, though, the View Source option was not buried in a developer's edition but was part of the edition everyone used, which encouraged even neophyte users to view the source of pages, whereupon they would see the relative simplicity of (especially early) HTML. (An interesting discussion about the need for the View Source option can be found in this bug report: https://bugzilla.mozilla.org/show_bug.cgi?id=256213 – which was a request to move View Source into a developer build of Firefox, and was ultimately rejected.)

This same expectation - that users should be able to view the raw source of files served by Web servers in addition to the rendered effect - was later extended to Cascading Style Sheets (.css files) and JavaScript (.js). In order to be rendered and displayed, browsers would need to download HTML, CSS, and JavaScript files, along with images and other binary files referenced in pages. However, the fact that the major browser developers chose to expose access to raw source as a first-level menu item was extraordinary. (In some cases accessing .css and .js files required a bit more ingenuity on the user's part, but nothing like the difficulty of accessing the source files in any other format such as Microsoft's Word or Adobe's PDF.)

The View Source option was, however, perfectly consistent with the culture of the Web as a whole. Certainly, as I've said above, this was consistent with Tim Berners-Lee's vision of the information Web in which all could participate, and in the choice of HTML as an immensely simplified version of SGML. One could argue that the View Source option simply reinforced the already "open" nature of the Web as a phenomenon, which was firmly

in place before the first browser was ever made widely available.

Regardless of whether the View Source option set the tone for the culture or merely reflected it, it is fair to characterize the culture of Web development that came into being as a View Source culture. In the early years of the Web, people learned HTML by example. When you saw a site that was doing something interesting, you would view the source of that site, and (often quite directly) copy their code into your own pages and edit from there. It became quite customary for such "imitation" (which was in some cases arguably copyright infringement by modern definitions) to occur even without the polite inquiry that originally preceded it. At first people asked, "Can I borrow your Web page template? I love what you've done with tables," but over time it became so customary people didn't even ask. People did continue to ask for access to server-side features, like Perl scripts for the common gateway interface, but perhaps due to Perl's heritage (version 3.0, in 1989 had been under GPL, even before Linux's public release), such scripts were commonly shared as well.

As the Web bubble grew, people started to offer well-designed HTML/CSS templates; useful JavaScript snippets; cascading drop-down, slide-out menu frameworks; scrolling marquees; and the like. Some of these libraries and templates were offered as freeware or shareware; some were sold as commercial software. Of course, developers couldn't easily prevent end users with the necessary JavaScript, CSS, and HTML in their browser cache from "borrowing" their scripts – it was

very common for such developers to find unlicensed versions of their code on Web pages from around the globe - but people had begun to realize that there was some value in the effort involved in making a useful function work as a cross-browser. In fact, one of the purported advantages of Java applets and Flash movies, when they first arrived, was the fact that they closed this loophole and were emphatically not View Source enabled.

Early efforts, however, to leverage JavaScript and CSS to provide a more compelling browser experience – what we then called DHTML – were hampered by the complexity of developing for multiple browsers, each of which had their own interpretation of the Document Object Model and implementation of Cascading Style Sheets. Two major events provided a way out of this complexity: the recognition of a community of practices now referred to as AJAX (itself enabled by the release of compelling competitors to Microsoft's Internet Explorer) and the appearance of a number of open source frameworks for developing AJAX applications.

AJAX

In what is now a seminal article, Jesse James Garrett noted, in February of 2005, a growing phenomenon of using standards-based presentation (XHTML and CSS), the Document Object Model (DOM), the XMLHttpRequest, and JavaScript to generate rich interface Web-based applications. He called this approach AJAX, and issued a call to designers and developers "to forget what we think we know about the limitations of the Web, and begin to imagine a wider, richer range of possibilities" ("AJAX: A New Approach to Web Applications" by Jesse James Garrett).

While many of the raw materials of the AJAX approach had been available for many years – Microsoft first introduced the XMLHttpRequest object in Internet Explorer 5, and it was leveraged in production by Outlook Web Access – it only took off when the approach was validated by inclusion in the Mozilla framework (and thus Firefox) and Apple's Safari browser (itself based on KHTML, the framework used for Konqueror in the K Desktop Environment for Linux). So long as what Microsoft called "remote scripting" was only available in a single browser (Internet Explorer), many developers felt it was unusable in any public application. (It might be acceptable, for some, to dictate browser usage on a corporate intranet, but most were unwilling to try to do so on the public Internet.)

As the user base of the old Netscape 4.x applications waned, and the availability of new

platforms rose, developers were emboldened by the relative ease of developing cross-browser standard applications with rich interaction in JavaScript and CSS, and more and more developers started to create AJAX applications. Garrett's essay crystallized this movement and gave it a name. The AJAX meme spread like wildfire, and the essay became so influential, exactly because so much work of this kind was already being done.

Open Source Frameworks

In addition to the AJAX phenomenon, the other major change that supported the new explosion of activity around Web development was the emergence of open source frameworks for AJAX-style development. While the diminishing use of Netscape 4.x (specifically, the acceptance among developers and their clients of the decision to not include complete functionality in Netscape 4.x as a primary requirement) and the relative stability of standards for HTML, CSS, and JavaScript all combined to make developing cross-browser Web applications considerably easier, writing such applications from scratch still required too much repetitive effort. Though the Mozilla-based platforms had adopted an approach similar to Microsoft's remote scripting, there were enough differences to create a barrier to cost-effective and reproducible development.

The frameworks that evolved abstracted away the messy details (the difference between Internet Explorer's ActiveX-based XMLHttpRequest object and Mozilla's true JavaScript version, for example) and enabled developers to add snazzy AJAX functionality to their Web applications simply by leveraging APIs provided by the framework.

Open Source Culture

In addition to ease of use, broad adoption of these frameworks has brought along with it an open source culture, characterized by an emphasis on community, de jure licensing, and more sophisticated planning and architecture for the presentation-tier of Web applications. (While the definition of open source typically focuses specifically on what licenses qualify, these licenses carry with them a set of cultural impacts, just as the View Source command brought with it a set of cultural impacts.)

First, and most obviously, an open source culture is characterized by explicit license rather than borderline theft or gift of use. While the View Source culture sometimes included explicit permission ("Can I copy what you've done here?"), it more often relied on a kind of borrowing that ranged from imitation

Framework	License	Notes
Prototype.js	MIT Style (slightly modified)	Server-Side Platform independent Fully integrated into the Ruby on Rails framework as well as Symfony framework for PHP
Script.aculo.us	MIT	Builds on Prototype.js
Rico	Apache 2.0	Builds on Prototype.js
Dojo	Dual: Academic Free 2.1 and BSD	Server-Side Platform independent Dojo foundation is supported by IBM, OpenLaszlo, AOL, JotSpot and others
YUI (Yahoo User Interface Elements)	BSD	Utilities and Controls (Widgets) in JavaScript, css
ZimbraTk	Dual: Mozilla Public or Apache	Client Developer Library – Widgets, Controls in JavaScript

Table 1: Open Source AJAX Frameworks

to outright copyright infringement or theft. An open source culture relies not only on a kind of de facto or assumption-based sharing of information, but a formally stated, de jure grant of specific rights to all users. This explicit license also facilitates innovation, as improvements made by users of the code can be contributed back into the community and the benefits of those innovations shared with other users directly.

Additionally, an open source culture is characterized by communities that develop around the frameworks. These communities have differing levels of formality, professionalism, and "helpfulness," but all represent a great leap forward compared to the random deciphering of other people's code, which characterized the View Source culture. To revisit Tim O'Reilly's metaphor about "View Source," in an open source culture, a would-be developer cannot only look over the shoulder of other developers, but he or she can join them in a conference room and discuss the code directly. Open source culture also helps to ensure repeatability and maintenance of the leveraged code base because when newer versions or patches are released, there is a well-organized mechanism for announcing and providing access to the code, something that the View Source culture generally lacked.

Finally, an open source culture is characterized by a greater degree of attention to standards and interoperability. Because open source projects gain their strength from the breadth of their use and the size of their community, they tend to focus much more clearly on methods of sharing. Even in cases where an individual's needs are contrary to the direction of the overall project, open source licensing allows for and encourages the development of extensions and alternate versions to meet specific problem sets.

Community and Professionalism

In some ways, one could point out the essential difference between the Web development culture, which relies heavily on open source frameworks and libraries, and the first generation of Web development, which relied on View Source and imitation, in the same way that free / open source software advocates have long distinguished between them: by drawing the "free as in beer" versus "free as in speech" (or "free as in freedom") comparison.

Viewing the source of a Web page developed by someone else, including digging into the CSS and JavaScript that accompanies the HTML, has always been free, as in beer. If you have access to a Web browser and can display the page, you can get to the source. Open source frameworks like Prototype.js, Dojo, DWR, the YUI library, and the Zimbra toolkit, however, are free as in speech. Not only can you access the source code without cost, you are encouraged and explicitly granted the right to use, modify, and redistribute your modifications to others.

The early days of the World Wide Web were characterized by open experimentation and the de facto sharing of source, whereas the trend today is toward maturity, professionalism, and the de jure sharing of open source frameworks. This evolution is a sign of the growing professionalism of the Web development community.

While the current AJAX-style development community faces many ongoing challenges - working out JavaScript namespace issues, encouraging better adherence to security best practices, and dealing with accessibility issues are at the top of a substantial list – the broad, active, and transparent communities behind open source AJAX frameworks bode well for the possibilities of solving such challenges and continuing the evolution of Web development. ⭕
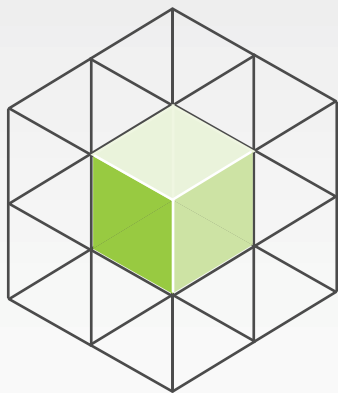
Open Management
Consortium

Systems Management is Now Open

In May 2006 the Open Management Consortium was announced to help advance the promotion, adoption, development and integration of open source systems /network management software. The founding members of the consortium are Ayamon, Emu Software, Qlusters, Symbiot, Webmin, and Zenoss.

Specific objectives of the Open Management Consortium include:

- Create awareness of open source management tools in the market
- Provide education and resources to help end users make informed decisions regarding open source
- Establish conventions and standards that enable integration and interoperability
- Enable collaboration and coordination on common development projects
- Promote collaborative open source systems management solutions

Open source systems management replaces monolithic vendor lock-in with a modular approach. Pick what you need, customize it to your exact specifications and add to it as your needs change. Because open source products welcome contributions by users, partners and other third parties, they must be standards-based. This standards-based approach facilitates interoperability between open source solutions across the systems management life cycle.

www.open-management.org

I CAN'T GET MY DATA OUT OF OUR CRM SYSTEM.

CUSTOMIZING OUR CRM SYSTEM IS OFF-LIMITS.

OUR CRM SYSTEM IS SO INFLEXIBLE.

SugarCRM® solves the problems that proprietary CRM solutions created.
Gain control over your business critical data.

Start using the most powerful commercial open source CRM solution for managing
sales, marketing, support and group collaboration activities today.

Get hooked on SugarCRM. Visit www.sugarcrm.com/eosm or call +1 408.454.6941.

Commercial Open Source Customer Relationship Management

SUGARCRM®
COMMERCIAL OPEN SOURCE
www.sugarcrm.com • www.sugarforge.org
1.87SUGARCRM • +1.408.454.6941