jQuery | Google Web Toolkit | Dojo | Ruby on Rails | Samba

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Co

FEBRUARY 2009 | ISSUE 178

REVIEWED:

» DELL INSPIRON MINI 9

» ARCHOS 5

DEVELOPING WEB SITES and VIRTUAL WORLDS

SECOND LIFE UNIVERSE ON OPEN-SOURCE CODE

Check Out jQuery
an Open-Source JavaScript Toolkit

Google Web Toolkit
Modern Web 2.0 Development without JavaScript

HOW-TO
Create Speedy Stats from Log Files

PLUS
Improving Ruby on Rails Default Web Pages with ActiveScaffold

Using Dojo's gfx Library for 2-D Drawing Tools

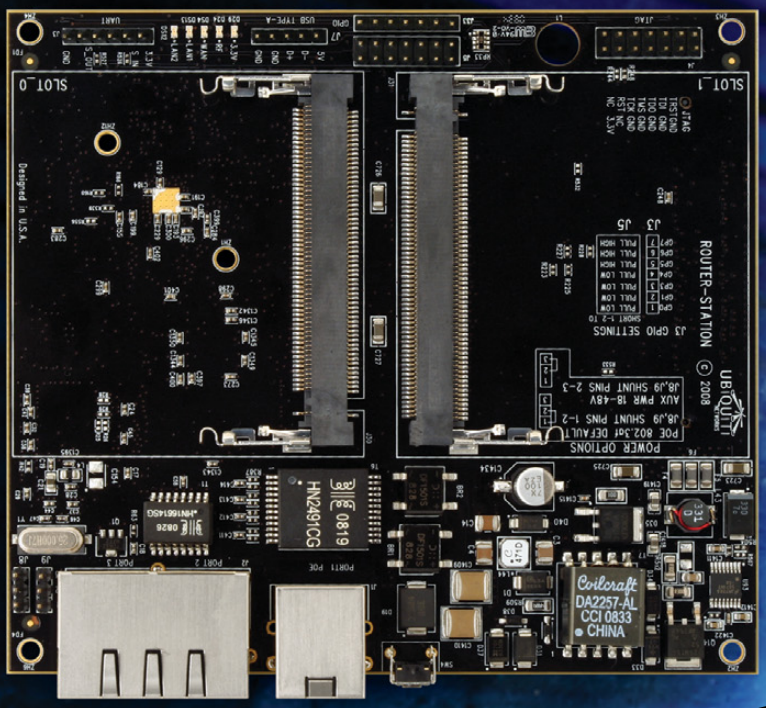
www.linuxjournal.com
$5.99US $5.99CAN
02
0 09281 03102 4

UBIQUITI NETWORKS

The Embedded Wireless Dream Machine.

680MHz CPU, 64MB RAM, 16MB Flash, USB, 3 mini-PCI slots, 5A power supply for multiple hi-power radios, and enhanced temperature performance and ESD protection for carrier applications. Full Linux SDK support and a $59 USD MSRP.

RouterStation



RouterStation UI CHALLENGE

CASH PRIZE

$200000

www.ubnt.com/challenge

www.ubnt.com

RouterStation UI Challenge: For Contest Details Visit: www.ubnt.com/challenge
Cash Prize $200000 (Two hundred Thousand) USD.

User Interface development based on OpenWrt Linux firmware. OpenWrt is a registered trademark of OpenWrt.

AMD Opteron 64

THE QUIET REVOLUTION

32 processor performance in your desktop

HiPerSTATION 8000

System Highlights:

- Optimal multi-threaded application performance utilizing (8) eight native Quad-Core AMD Opteron™ processors
- Up to 256GB of DDR2-667 memory
- Next generation dual x16 PCIe technology offers scalable graphic bandwidth
- 2TB of storage
- Designed to be quiet enough to utilize in an office environment; ⅓ the noise of server version
- Ideal for usage in Computer Aided Designs, Visualization, and Digital Content Creation

HPC SYSTEMS INCORPORATED

www.hpcsystems.com 888-725-3472

AMD Solution Provider PLATINUM

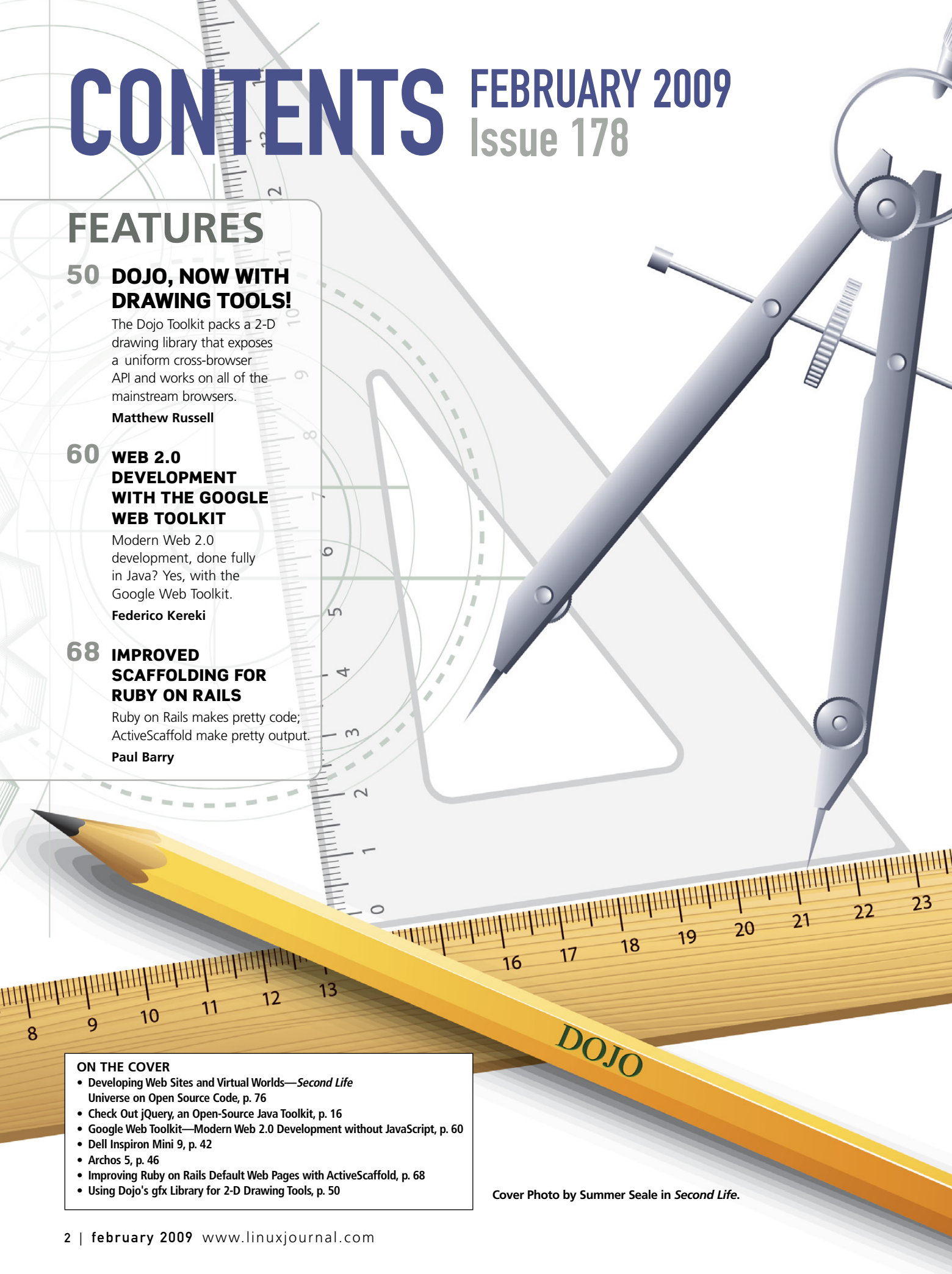
GSA GS-35F-0596R Contract Holder

SBA 8(A) CERTIFIED

CONTENTS

FEBRUARY 2009
Issue 178

FEATURES



ON THE COVER

Cover Photo by Summer Seale in *Second Life*.

Looking for a domain?

More ways to get the domain name you REALLY want!

New!

Register new domains ...

When you search for a domain name, our domain checker automatically gives you relevant domain suggestions, ensuring that you find the best domain name for your website.

... or buy a domain that has already been registered.

If the domain name you want has already been registered, you can use our FREE domain auction service to search over 14 million domain names and make a bid on the one you want.

FREE Private Domain Registration

Your privacy is important to us. With free private domain registration, your name, address and phone number remain confidential. Stop spammers, telemarketers, data miners and anyone else who might misuse your personal data.

Save 25% on .com domains!*

*Offer valid for a limited time only. Discount applied to first year of a 2 year minimum contract term. Visit www.1and1.com for full promotional offer details. Product and program specifications, availability and prices subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet AG, all other trademarks are the property of their respective owners.

MEMBER OF united internet

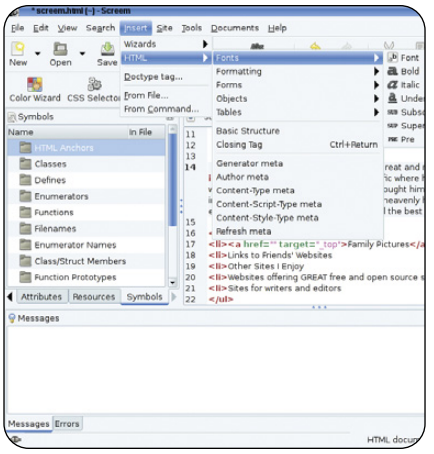
Call **1-877-GO-1AND1**

Visit us now **www.1and1.com**

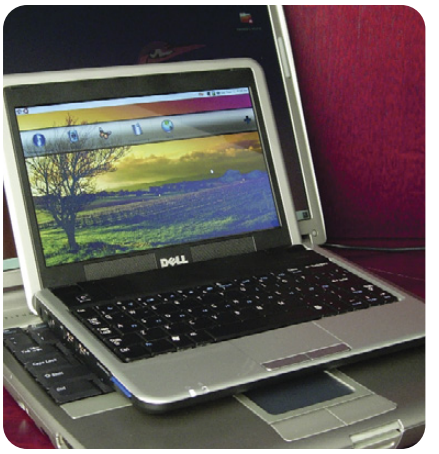
1&1

CONTENTS FEBRUARY 2009 Issue 178

76 *SECOND LIFE*



20 SCREEM



42 DELL INSPIRON MINI 9

COLUMNS

INDEPTH

REVIEWS

IN EVERY ISSUE

Next Month

DESKTOP

Linux has come a long way since we saw that first shell prompt, even if we still love using the command line. Now we have GUI desktops with stunning graphics and a multitude of GUI applications.

If you thought you had only two GUI desktop choices, you need to read our article on Xfce. If you're a GNOME user, learn how to use your desktop more efficiently with GNOME Do. Or, are you looking for a browser alternative for your desktop? If so, read our review of three alternative browsers (alternatives to Firefox, of course).

And, just when you were getting comfy with that desktop, read about Adobe Air on Linux and see how it may make your desktop irrelevant.

USPS *LINUX JOURNAL* (ISSN 1075-3583) (USPS 12854) is published monthly by Belltown Media, Inc., 2211 Norfolk, Ste 514, Houston, TX 77098 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is $5.99 US. Subscription rate is $29.50/year in the United States, $39.50 in Canada and Mexico, $69.50 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 16476, North Hollywood, CA 91615. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

Need a server for your small business?

Look no further.

Introducing 1&1 Premium Servers, a powerful new line of dedicated servers specifically designed for high performance needs. Featuring energy efficient technology, these top-of-the-line machines reduce costs and environmental impact with increased performance-per-watt.

AMD Quad-Core Server

AMD Opteron 64

THIS COMPANY USES CERTIFIED CLEAN ENERGY
b-e-f.org

Save BIG in 2009!

All 1&1 Premium Servers include*:

- RAID
- Plesk 8 with 100-domain license
- FTP backup space
- Serial Console: Connect directly to the serial port of your server, even if the network is down.
- Recovery Tool: Load rescue image and reboot server.
- GeoTrust Dedicated SSL certificate (a $49 value!)
- Easy-to-configure firewall and more!

1&1 BUSINESS SERVER II

Dual-Core AMD Opteron™ 1218, 2 x 2.6 GHz,
4 GB DDR RAM, 2 x 500 GB Hard Drive

~~$199.99~~ per month — 3 months FREE!*

1&1 ENTERPRISE SERVER I

Quad-Core AMD Opteron™ 1352, 4 x 2.1 GHz,
4 GB DDR RAM, 2 x 750 GB Hard Drive

~~$299.99~~ per month — 3 months FREE!*

1&1 ENTERPRISE SERVER II

Quad-Core AMD Opteron™ 1356, 4 x 2.3 GHz,
8 GB DDR RAM, 2 x 1,000 GB Hard Drive

~~$399.99~~ per month — 3 months FREE!*

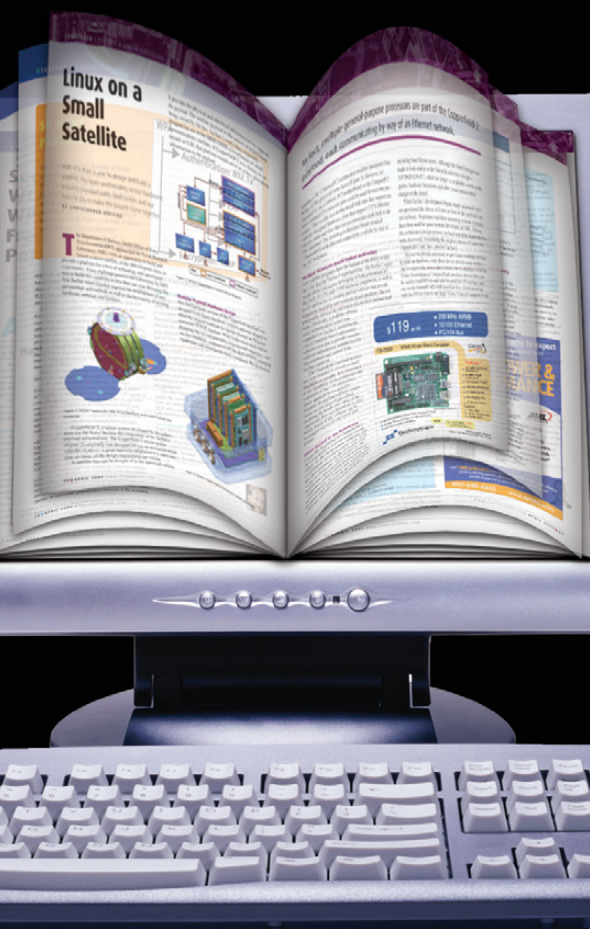
*Offer valid for a limited time only. Prices based on Linux servers. Discount applied to first 3 months of a 12 month minimum contract term, $99 setup fee applies. See www.1and1.com for full promotional offer details. Certain features not available with Managed Servers. Product and program specifications, availability and prices subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet AG, all other trademarks are the property of their respective owners.

MEMBER OF united internet

Call **1-877-GO-1AND1**

Visit us now **www.1and1.com**

1&1



LINUX JOURNAL

Advertising
E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions
E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
PHONE: +1 818-487-2089
FAX: +1 818-487-4550
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 16476, North Hollywood, CA 91615-9911 USA
Please allow 4–6 weeks for processing address changes and orders
PRINTED IN USA

servers DIRECT®

MORE PRODUCTS, BETTER SERVICE, GUARANTEED.

GO STRAIGHT TO THE SOURCE!

1.877.727.7887 | www.ServersDirect.com

YOUR HIGH PERFORMANCE COMPUTING HAS ARRIVED.

The ServersDirect® Systems with the Intel® Xeon® Processor helps you simplify computing operations, accelerate performance and accomplish more in less time

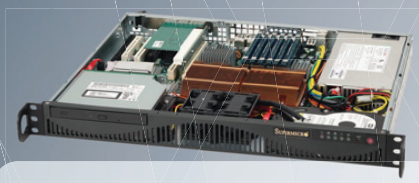
INTEL® MODULAR SERVER

The Intel Modular Server is an integrated system built on Intel Multi-Flex Technology that includes SAN storage, computing and networking to simplify the growing demands of your IT infrastructure.

The Intel Modular Server features:

- Up to six server Compute Modules
- Integrated SAN storage
- Integrated Ethernet Switch Modules
- All hot-swappable components
- Virtual Presence remote management capabilities

For more information visit: http://www.ServersDirect.com/IMS.html



SDR-S1208-T00

STARTING AT **$559**

- Supermicro Mini 1U Rackmount Server with 260W Power Supply
- Supermicro Server Board w/Intel® 946GZ Chipset
- Support up to a Dual-Core Intel® Xeon® 3000 Series processor
- TPM Support
- 1x 3.5" Internal Drive Bay
- 2x Intel® 82573 PCI-e Gigabit LAN Port

SDR-C1303-T02

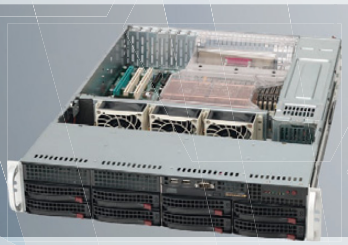
STARTING AT **$899**

- 1U Rackmount Server with 400W Power Supply
- Supermicro Server Board w/ Intel® 5100 Chipset
- Support Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 48GB 667/533MHZ DDR2 ECC Reg
- 2x 3.5" Hot-swap SATA Drive Bay
- Intel® 82573V & 82573L Gigabit Ethernet Controller

SDR-C2301-T06

STARTING AT **$1,059**

- 2U Rackmount Server with 460W Power Supply
- Supermicro Server Board w/ Intel® 5100 Chipset
- Support Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 48GB 667/533MHZ DDR2 ECC Reg
- 6x 3.5" Hot-swap SATA Drive Bay
- Intel® 82573V & 82573L Gigabit Ethernet Controller



SDR-S2301-T08

STARTING AT **$1,499**

- 2U Rackmount Server w/700W High-Efficiency Redundant Power Supply
- Supermicro Server Board w/Intel® 5000P chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 64GB DDR2 667 & 533 FB-DIMM
- 8 x 3.5" Hot-swap Drives Trays
- Dual-port Gigabit Ethernet Controller

SDR-C3301-T16

STARTING AT **$1,399**

- 3U Rackmount Server with 650W Power Supply
- Supermicro Server Board w/ Intel® 5100 Chipset
- Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 48GB 667/533MHZ DDR2 ECC Reg
- 16 x 3.5" Hot-swap SATA Drives Trays
- Dual-port Gigabit Ethernet Controller

SDR-C4302-T02

STARTING AT **$899**

- 4U Rackmount Server with 600W Power Supply
- Supermicro Server Board w/ Intel® 5100 Chipset
- Support Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core
- Support up to 48GB 667/533MHZ DDR2 ECC Reg
- 3 x 3.5"Internal SATA Drives Trays
- Dual-port Gigabit Ethernet Controller

SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL US TODAY!

Our flexible on-line products configurator allows you to source a custom solution, or call and our product experts are standing by to help you assemble systems that require a little extra. Servers Direct - your direct source for scalable, cost effective server solutions.

1.877.727.7887 | www.ServersDirect.com

intel Xeon® inside™

Dual-core. Do more.

SHAWN POWERS

Making the Web a Little Less Sticky

Not too terribly long ago, Web development literally meant learning how to write hypertext markup language code that would format text and graphics properly when viewed on a Web browser. We really didn't have many choices, nor did we have very many ways to solve a problem. If you wanted your text italicized, you surrounded it with <i> tags. (We won't go into the <i> versus debates that still live on today.) The Web, however, is no longer just a one-way information-dissemination tool. The World Wide Web is interactive, and it's actually an application platform that holds the promise of obscuring the underlying operating system out of existence. And, what runs the back end for most of those Web applications? Yep. Linux.

On most levels, programming for the Web is no different from programming for an operating system. In fact, many of the same languages lend themselves quite nicely to the Web world. We've devoted this issue to help sort out some of the options. Whether you're an old hand looking for a few new ideas to optimize your Web apps or someone new to Web development looking for the right tool for the job, we think you will enjoy this issue.

Reuven Lerner gives us all a lesson in jQuery. It's growing in popularity, and Reuven shows us some reasons why. It certainly doesn't mean you have to switch if you're already using something like Prototype, but it's definitely something you'll want to read about. Although, perhaps something like the Google Web Toolkit is more appealing. There's no doubt Google knows its stuff when it comes to Web development, and using the GWT, you can harness much of Google's power from within your Java programs. Federico Kereki walks us through developing Web 2.0 applications using Google's Web toolkit, again emphasizing the idea of the Web as a "platform" rather than just a device for passing data.

One of the beauties of new Web technologies is that not only can we create text dynamically, but we're to the point where creating dynamic graphics is possible too. Matthew Russell shows us how to do just that using Dojo and JavaScript. Gone are the days of static-only graphics on Web sites.

I want to let you in on a little secret. A few paragraphs up, when I mentioned <i> versus , sadly that about summed up my Web programming abilities. If you're in the same boat I am, fear not; we made sure to keep this issue relevant to you as well. Marcel Gagné highlights a handful of HTML editors that make it easy, even for nonprogrammers, to create Web pages. For many of us, that still suffices.

If you aren't interested in creating Webby goodness at all, every issue of *Linux Journal* is designed to appeal to all our readers. Bill Childers introduces us to an entire virtual on-line world with *Second Life* in Linux. When you add Mick Bauer's series on Samba Security, Dave Taylor's shell scripting, Daniel Bartholomew's reviews of the Dell Mini 9 and the Archos 5, and Kyle Rankin's tutorial on hacking apart log files, I'm not sure how we fit everything between the covers!

This is a fun issue of *Linux Journal*, and I think it will appeal to a wide variety of readers. Whether you're a Web programmer or a gamer, a Ruby on Rails fan or a Netbook enthusiast, it will be a good month. <i> have a feeling that you'll all <b> impressed with most of the topics covered this month, and hopefully some of you will enjoy all!■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

zt Systems

ZT Systems delivers something different: a unique

B A L A N C E

of world-class server performance and cost advantage joined with extensive flexibility in delivery and support

From the small office to the mega-datacenter, ZT Systems provides dependable, energy efficient server and client solutions that maximize productivity and optimize your IT budget. An established manufacturer celebrating our 15th anniversary, ZT combines individualized attention to your requirements with the experience and scalability you demand. All ZT Systems are assembled and serviced in the USA.

Choose ZT Systems and get:

- Custom Configurations
- Professional Grade Quality
- Personalized Account Service
- Superior Value

ZT Reliant SC1212i
Small Business Server
Entry-level Single Socket Solution

Starting at Only $849

ZT-SC1212i-35

- Intel® Core™2 Quad processor Q6600
- 4GB DDR2 800MHz SDRAM (2x2GB)
- Intel® 3200 Server Chipset
- (2) 500GB SATAII 32MB Hotswap Hard Drives (Max.4)
- RedHat Enterprise 5 Trial Version
- 3 Year Limited Warranty and 24x7 Telephone support

ZT Systems 1104Ri
1U Rack Server
Quad-core 1U server with 8 hard drives

Starting at Only $1349

ZT-1104Ri-82-C00001

- Intel® Xeon® Processor X3330
- 4GB DDR2 667MHz Unbuffered SDRAM (4x2GB)
- Intel® 3200 Server Chipset
- (2) 150GB 2.5" SATA 10Krpm Hotswap Hard Drives (Max. 8)
- RedHat Enterprise 5 Trial Version
- 3 Year Limited Warranty and 24x7 Telephone support

ZT Systems 1205Ri
1U Rack Server
Dual-socket Data Center Server

Starting at Only $1599

ZT-1205Ri-84-C00001

- (2) Intel® Xeon® Processors E5410
- 8GB DDR2 ECC/REG. SDRAM (4x2GB)
- Intel® 5100 Server Chipset
- (2) 500GB SATAII 32MB Hotswap Hard Drives (Max. 4)
- RedHat Enterprise 5 Trial Version
- 3 Year Limited Warranty and 24x7 Telephone support

ZT Systems 4201Ci-84
4U/Tower Rack Server
Convertible Tower / Rack SMB Solution

Starting at Only $1999

ZT-4201Ci-84-C00002

- (2) Intel® Xeon® Processors X5410
- 8GB DDR2 ECC/REG. SDRAM (4x2GB, Dual Rank)
- Intel® 5100 Server Chipset
- (2) 1TB SATAII 32MB Hotswap Hard Drives (Max. 8)
- RedHat Enterprise 5 Trial Version
- 3 Year Limited Warranty and 24x7 Telephone support

Scalable Custom Server Solutions - Contact Us to Learn More

(866) 984-7687 corpsales@ztsystems.com

letters

Satisfaction

After having used the internal-built G technologies with my Acer Aspire 3690, I must say with great satisfaction that the Dynex Draft N card is a solid performer. Having used Ubuntu for two years, I'm glad to see that wireless speeds are surpassing those of Windows. Where I live, everyone is wired into Windows, and as an Ubuntu fan for two years, it's amazing to have my laptop running faster than most Windows machines—as well as without the crashes all too common with Vista, my 250GB hard drive automatically working, my 2GB of RAM not conflicting with the system, and my Draft N card pulling its weight. Ubuntu, by far, is one of my favorite operating systems. I'm praying that sooner or later, all laptops and PCs are given the options to have either Windows or Linux as their primary OS.

--

Joseph Ziehm

It's Happening

I'm behind in my reading and just finished "Linux for the Long Haul" by Michael Surran [*LJ*, August 2008], about the Houlton Christian Academy's migration from Windows. Like most businesses using Linux, "GHCA has a single Windows machine in our office for the sole purpose of running Intuit's QuickBooks". Intuit has *finally* begun to realize that its future is not on the Windows desktop. A version QuickBooks Online that is compatible with Firefox (and other non-IE browsers) is in the works. Now, if we can just get Photoshop....

--

Joe Holt

Liked That Tech Tip

I really liked the Tech Tip on page 56 of the December 2008 issue. Being a bit of a bug for efficiency, I will mention one possible improvement. However, it may work only with the Bash shell. I am not very familiar with the other shells. I do this sort of thing because Ben Franklin once said: "A cycle saved is a cycle earned!" Or something like that. I've worked on some really slow machines in my day.

The line:

```
F=$(echo $F| perl -pe 's/.gz$//')
```

could be replaced with the line:

```
F=${1%.gz}
```

which allows the line:

```
F=$1
```

to be eliminated entirely.

And, just because I like to be different, I think that the line:

```
nice gunzip -c $F
```

would "look better" if gunzip were replaced with zcat. I think that zcat is simply more "intuitive" than `gunzip -c`:

```
nice zcat "$F"
```

Also notice that I enclosed $F in double quotes just in case there might be one or more blanks in the filename, which would make the unquoted $F look like multiple arguments to the zcat.

Oh, and just as a question, would sed be more efficient than perl here:

```
F=$(echo "$F"|sed 's/.gz$//')
```

Just curious on this last one.

Again, many thanks for the tip.

--

John McKown

Really, Really Liked That Tip

I really did like that tip [see letter above]. Using the idea in it, I created the following two functions that I now

PHOTO OF THE MONTH

Have a photo you'd like to share with *LJ* readers? Send your submission to publisher@linuxjournal.com. If we run yours in the magazine, we'll send you a free T-shirt.

Logan Bryngelson, submitted by his father, Ryan Bryngelson.

sourced by my Bash profile:

```
function do_cat()
{
 local CAT
 case "$1" in
 *.gz) CAT=zcat;;
 *.bz2) CAT=bzcat;;
 *) CAT=cat;;
 esac
 $(${CAT} "$1")
)

function smart_cat()
{
 local i
 for i in "$@"; do
 do_cat "$i"
 done
}
```

Very thought-provoking tip! Of course, the do_cat function can be extended to handle other cat-like commands simply by including more entries in the case portion of the do_cat() function. I guess I could have created only a single function of smart_cat(), but I like the separation of using two functions.

--

John McKown

Correction

In the December 2008 issue of *Linux Journal*, the "Going MoBile" interview said that *Linux Journal*'s mobile site, **m.linuxjournal.com**, ran on Linux-based MoFuse. Instead, it runs on Drupal (as does our main site), using a theme optimized for mobile devices.

--

Doc Searls

***LJ* pays $100 for tech tips we publish. Send your tip and contact information to techtips@linuxjournal.com.**

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

At the most recent kernel summit, **Greg Kroah-Hartman** and a variety of other kernel folks decided to create the **Staging Subsystem**. It's not really a subsystem in the normal sense of the word. It has no particular technical focus. Instead, the Staging Subsystem is a place in the kernel source tree where new coding projects can go, before they are really ready to be included in the kernel proper. It's a way to get code into public use quickly, by including it in an isolated part of the tree. All kinds of crazy stuff can go in there. Greg has listed himself in the MAINTAINERS file as the official maintainer, but his task will be quite different from traditional code maintenance. He plans to make sure submissions actually compile, but beyond that, there will be a wide array of breakage, ugliness and general chaos in the Staging Subsystem. This is all according to plan.

The overall motivation for inflicting this kind of mess on the kernel is to accommodate **Linus Torvalds**' new efforts at accepting code more quickly into the main tree. Because Linus intends to take code that has been tested less thoroughly, and perhaps written less carefully, into the kernel, Greg and others want to give that code every chance to be tested and improved before Linus accepts it. All of this should result in an ever-increasing speed of kernel development that ultimately probably was made possible by the transition from BitKeeper to git many moons ago.

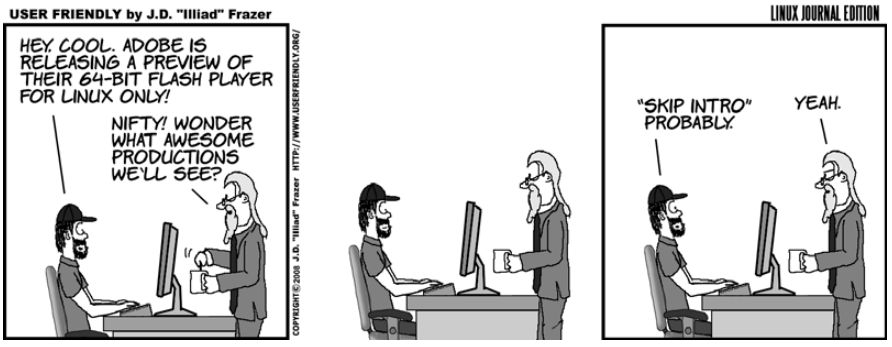
Linus also has expressed a desire to change the way **kernel versions** are numbered. Ever since he decided to throw over the old paradigm of "x.even is for stable series", and "x.odd is for development series", the kernel has remained at version 2.6.x, with "x" just getting bigger and bigger. With no clear threshold for incrementing either the 2 or the 6, Linus, Greg and others are having a harder time dealing with all the versions. Imagine looking at a dozen different version numbers each day. They start to blur together. Linus asked folks to consider alternative version-numbering schemes that would be easier to read and just better in general. Unfortunately, he didn't give more clarity than that, and when Greg opened up a discussion on the mailing list, things degenerated into bickering.

The issue of the version-numbering scheme is closely related to Linus' new plan of accepting code more quickly into the kernel and to keep development as active as possible. When he abandoned the old even/odd system, the old version numbering stayed in place, even though it no longer meant anything. Now it seems clear that a new versioning system will be put in place in the near-to-medium term. But, judging from the discussions that already have taken place on the issue, it may be a while before any meaningful suggestion comes up. A number of developers, including **Alan Cox**, think any system will have drawbacks, and so keeping the current one ultimately will be best. But Linus wants a change, so it's very likely that he'll pick a new number scheme before too long.

Efforts to eradicate the **Big Kernel Lock** (BKL) continue apace. **Frederic Weisbecker** has come up with a tool to help folks target which cases of the BKL should be eliminated first. Frederic's tool, the **Big Kernel Lock Tracer**, tells how much time is used by each instance of the BKL over the course of a system's uptime. Instances of the BKL that use the most time presumably would be the best targets for replacement by simpler locking structures. The BKL itself has proven to be remarkably difficult to get rid of. Because it's so ham-handed in its approach to resource locking, no other parts of the system can do anything while a piece of code holds the BKL. Linus has wanted to replace it with simpler locking structures for a long time, which would allow the rest of the system to continue to function and lock up access only to the specific resources in question. But the BKL is so ubiquitous in the kernel, and the locking requirements for much of its usage are so complex and individually nuanced for each situation where it occurs, that a straightforward replacement is often just impossible. In recent months, however, efforts to remove the BKL have become more intense, and things like Frederic's tool only increase that intensity.

Representing the **load average** of a running system as just a single number has some drawbacks—what if you want to know the networking load average as opposed to the disk usage load average? **Sena Seneviratne** and **David Levy** have been working on a mechanism that would separate out those different kinds of load averages and make those calculations available to the user at runtime. It's possible that the work they're doing may make it into the kernel at some point, but as **Arjan van de Ven** points out, it's important not to break user space. All the existing tools that rely on the current meaning of the load average must continue to work. His suggestion to Sena and David is that they feel free to add as many load average stats as they please, so long as they keep the original total load average as well.

—ZACK BROWN



LJ Index, February 2009

1. Billions of photos on Facebook by October 10, 2008: 10
2. Range in terabytes of photos uploaded to Facebook daily: 2–3
3. Billions of photo images served by Facebook daily: 15
4. Millions of merchants on the Amazon platform: 1.2
5. Billions of dollars spent on 2008 US presidential campaigns: 2.4
6. Billions of dollars spent on 2008 US congressional campaigns: 2.9
7. Billions of dollars spent by Americans in 2008 on potato chips: 10.6
8. Thickness in inches of a dollar bill: .0043
9. Stack height in feet of one million $1 bills: 358.33
10. Stack height in miles of one billion $1 bills: 67.86
11. Stack height in miles of 700 billion $1 bills: 47,506
12. Apache's percentage share among Netcraft's Top Developers in October 2008: 50.43
13. Number of Linux-based hosting companies among Netcraft's top ten most-reliable in September 2008: 4
14. Number of Linux-based hosting companies among Netcraft's top two most-reliable in September 2008: 2
15. Number of Linux-based hosting companies among Netcraft's top 50 most-reliable in September 2008: 24
16. Minimum number of Free the Penguin Multistation SUSE Linux Desktops deployed by November 2008: 20,000
17. Minimum number of schools receiving Free the Penguin Multistation SUSE Linux Desktops by November 2008: 3,000
18. Estimated millions of ASUS Eee PCs shipped in 2008: 6
19. Percentage of ASUS Eee PCs running Linux: 30
20. Millions of new ASUS Eee PCs for 2008 running Linux: 2.4

Sources: 1–3: Don Beaver of Facebook 4: Werner Vogels of Amazon | 5, 6: Center for Responsive Politics | 7: George Will in the *Washington Post* | 8–11: Betty Schier, the *News Review* (Roseburg, Oregon) 12–15: Netcraft.com | 16, 17: *Omni*, Userful and Novell | 18–20: *ComputerWorld*

Smarter Than Phones

The phone business is changing at a rate so fast, and on such a curved path, that Heisenberg's Uncertainty Principle (**en.wikipedia.org/wiki/Uncertainty_principle**) comes to mind. Where it is and where it's going may be conjugate variables, but trying to reconcile the two is kind of futile.

In November 2008, the research firm Canalys released its Q3 2008 report on "smartphones" (**www.canalys.com/pr/2008/r2008112.htm**). Worldwide sales were up 28%. Nokia still held the lead with a 38.9% share of shipments, but that was down 3.5% from a year earlier, and the number of phones shipped was down too. Apple meanwhile moved into second place with a 17.3% share, with unit numbers up 523% over the year before. And, that's with just two generations of a single phone—not a fleet of phones such as Nokia's...and everybody else's.

Among operating systems, Symbian was first and Apple second. Following were RIM, Microsoft and Linux, which had a 5.1% share and 49% growth. But that was before Android.

T-Mobile's Android phone hit only last October 17, 2008, early in Q4. Here's a telling quote from the report: "Motorola, currently holding onto fourth place in smartphones thanks largely to its Linux-based models, recently announced it would move away from using the Symbian OS and focus more on Android." Which is also Linux.

Both the iPhone and the Androids are platforms for running countless applications, only one of which is voice telephony. I know lots of people whose day-to-day digital lives are moving from their laptops to their iPhones, BlackBerries and, yes, Androids. Although the "war" between iPhones, BlackBerries and Androids will attract the most attention, all three will win the battle of computing over telephony in the mobile world.

Still, it's hard to do serious computing apps on networks built for routing calls and charging out minutes. It'll take longer for that battle to be won, but it'll happen too. The phone system will become a data system. It will be borged by the Net.

What happens next is up to developers. For more about that, see this month's EOF, "Net Development", on page 80.

—DOC SEARLS

Linux on the Label

Anyone that uses Linux regularly is familiar with the "Google to see if it works under Linux" procedure before buying any hardware. I was thrilled when I saw the ad for a USB Atari 2600 joystick clone that had a label on the box claiming its Linux compatibility. How cool!

—SHAWN POWERS

They Said It

The hungry and cold unemployed masses aren't going to continue giving away their intellectual labor on the Internet in the speculative hope that they might get some "back-end" revenue. "Free" doesn't fill anyone's belly; it doesn't warm anyone up.
—Andrew Keen, www.internetevolution.com/author.asp?section_id=556&doc_id=166342&

...this recession will be great for free and open source because of the shortage of cash. Last recession saw the mainstream legitimisation of open-source operating systems (youngsters, take note: there was a time when it wasn't automatically okay for an IT department to use Linux), because it was clear and away the most cost-effective choice. The saying I use is, "come for the price, stay for the quality". Perhaps this recession will legitimise many of the applications (CRM, finance, etc.) higher up the stack.
—Nat Torkington, radar.oreilly.com/2008/10/effect-of-the-depression-on-te.html

Since people contributing to open-source projects and on-line communities have not (unless paid to do so) been paid to do so, thinking that they'll stop just because they have potentially more time on their hands doesn't make much sense. Contributors to free stuff usually do it for reasons other than money.
—Rich Sands, rich-sands.com/wordpress/?p=10

Open is the new black.
—Marc Canter,
blog.broadbandmechanics.com/2008/11/open-is-the-new-black-continues-to-spread#respond

CERN's decision to make the Web foundations and protocols available on a royalty-free basis, and without additional impediments, was crucial to the Web's existence. Without this commitment, the enormous individual and corporate investment in Web technology simply would never have happened, and we wouldn't have the Web today.
—Tim Berners-Lee, tenyears-www.web.cern.ch/tenyears-www/Welcome.html

Linux Gets Faster with Splashtop

One of the nagging problems for Linux is that the most popular laptops are still codesigned by Microsoft and its OEMs. It's not for nothing that laptops come with stickers on the bottom that say, "Windows Vista—Business OEM Software" or whatever. These are not white boxes. You can get Linux running on them, but the hermit crab approach isn't the swiftest route to market leadership.

It's starting to look like that route may come through Splashtop, by DeviceVM. Splashtop starts a laptop in just a few seconds. Its Web site explains:

> Splashtop is preinstalled on the hard drive or in the onboard Flash memory of new PCs and motherboards by their manufacturers. Splashtop is a software-only solution that requires no additional hardware. A small component of Splashtop is embedded in the BIOS of the PC—that's the part that runs as soon as you press the power button.
>
> Within Splashtop, you have the choice of running one of its applications, such as the Splashtop Web Browser, or booting your operating system. Splashtop is compatible with any operating system, including Windows and Linux.
>
> Splashtop has similar networking capabilities to what you find in other operating systems. It can connect to networks over Wi-Fi, LAN, xDSL and cable. WEP, WPA and WPA2 wireless security standards are supported.

Note that first line. Splashtop does for Linux what those old OEM deals did for Microsoft: gives it a leg up, an advantage right out of the startup gate (pun intended).

At the time of this writing, Splashtop is preinstalled on laptops from ASUS, VoodooPC and Lenovo, and on all motherboards from ASUS. Every one of them is winning where it counts most with users—by saving time.

Splashtop is also committed to open source. At the time of this writing, it's still building its SDK. Check the Developers page at **www.splashtop.com** for progress on that. Meanwhile, expect to see more news about how Linux is winning the battle for quick startup times.

—DOC SEARLS

Who Will YOU Be at LinuxJournal.com?

Linux Journal readers are part of a pretty great community. Most of you like to come hang out with us over at LinuxJournal.com from time to time, and we'd like to make that experience a little more personal and fun. There will be some cool new features and ways to interact in the coming months, but to kick things off, we thought we'd encourage you to upload an avatar to represent yourself on the site. It can be a photo, graphic, text or anything you can dream up (as long as it is...ahem..."appropriate"). Look for some new features soon, and in the meantime, we look forward to seeing your new on-line identity. You can spot me on-line pretty easily now by my avatar too.

—KATHERINE DRUCKMAN

What They're Using

Mark Pilgrim and the Latest Essentials

Mark Pilgrim's original 2006 list of Linux Essentials (**diveintomark.org/archives/2008/10/28/essentials-2008**) was one of the inspirations for "What They're Using". For many, it comprised a handy shopping list of free stuff, graced by proper snarks in the general direction of proprietary alternatives. About Amarok, Mark wrote, "It's just like iTunes except it automatically fetches lyrics from Argentina, automatically looks up bands on Wikipedia, automatically identifies songs with MusicBrainz, and its developers are actively working on features that don't involve pushing DRM-infected crap down my throat."

Well, just in time for this month's "What They're Using", Mark came out with his Essentials, 2008 edition. With his permission, here is his posted text, verbatim:

> Essentials, 2008 edition
>
> Via email, "Chris" asks:
>
> Enjoyed the 2006 Linux essentials list....Do you have any plans to update the Essentials list for the latest and greatest that Linux 2008 has to offer?
>
> Well, I do now.
>
> 1. Debian GNU/Linux, because of Firefox bug 354622. In particular, comment 39 outlines Debian's position. (Yes, I know Debian still ships with nonfree firmware, so Debian's position on Firefox is inconsistent. But no firmware developer has ever tried to force Debian to "bend the DFSG a little").
>
> 2. GNU Emacs, still.
>
> 3. Iceweasel (see above) + Adblock Plus + NoScript + NoSquint + Greasemonkey.
>
> 4. Pidgin (formerly GAIM).
>
> 5. getmail instead of Thunderbird. It turns out I don't actually want a desktop e-mail client. I use Google Apps For Your Domain to manage my own address @ my own domain, and I archive it with a nightly cron job using getmail.
>
> 6. gPhoto 2 instead of digiKam. It turns out I don't actually want a photo library, just dated directories of image files imported directly from my camera.
>
> 7. On the other hand, I do want a music library, and Amarok is still best of breed.
>
> 8. KSnapshot, KTorrent, Konversation, k3b and k9copy are also best of breed.
>
> 9. There is still no good iMovie clone for Linux, but OpenMovieEditor looks promising. I learned about it at FSOSS last week. I fear that Linux video editing will always just "look promising".
>
> 10. I've warmed up to GIMP, which has largely replaced Krita for my (very light) graphic-editing needs. At FSOSS, I learned about ingimp, which allows you to opt in to collect and aggregate detailed statistics on how real people use GIMP.
>
> 11. GNOME Do, a Quicksilver-inspired launcher.
>
> 12. MPlayer, specifically the version provided by the awesome debian-multimedia.org.
>
> On the command-line side, I now use urxvt, screen, ZSH and these configuration files. If you use the command line more than once a day and haven't learned about screen yet, you're missing out. I still use SSH heavily, in too many ways to count. If you use SSH and haven't learned about SSH keys and SSH tunneling yet, you're really missing out. Also: sshfs, rsync and SSH, SSH VPN, &c. (Note: most of these work on Mac OS X too, and Windows with Cygwin or PuTTY.)
>
> I still use rsync for backups to my NAS, even though I have two Drobo enclosures that I manage with drobo-utils. RAID is not a backup solution, and ZFS on Linux isn't quite there yet.
>
> Things I don't use anymore:
>
> 1. A desktop e-mail client. As mentioned above, I use Gmail (on my own domain, so I keep my e-mail address).
>
> 2. OpenOffice.org, or any other desktop office suite. I use Google Docs, which exports to Open Document format. I keep studious local backups in ODF/ODS/etc.
>
> 3. VLC. There is very little that MPlayer can't handle.
>
> 4. Democracy Player. Ze frank doesn't post daily anymore, and I don't particularly follow other video podcasts.
>
> 5. KMyMoney. It's good, but I fell behind in organizing my finances and it got left on the wayside.
>
> 6. EasyUbuntu (now Medibuntu). Debian-multimedia.org satisfies all my illicit patent-encumbered needs.
>
> 7. Beagle. GNOME Do is more than enough for my local search needs.
>
> 8. Konsole. RXVT-Unicode instead.
>
> 9. AllTray.
>
> 10. Brightside.
>
> I also no longer use the ratpoison window manager. I've settled on XFCE instead, with the PCMan file manager. I encourage every Linux user to try an alternate window manager for at least a month. Find one that fits your brain and customize the hell out of it.

For linky goodness, go to Mark's source at **diveintomark.org/archives/2008/10/28/essentials-2008**.

—DOC SEARLS

jQuery

An initial look at jQuery, an increasingly popular JavaScript library.

REUVEN M. LERNER

JavaScript has become, somewhat surprisingly, one of the hottest programming languages around. The question is not whether Web developers need to know JavaScript, but rather what library they should use when working with it. That's because the core JavaScript language is a bit rough around the edges, with incompatibilities across different browsers and platforms. This is compounded somewhat by other cross-platform browser differences, such as various ways that event handling is implemented, which can make it hard for developers to deal with such problems.

Most of my Web development work in the past few years has been in the Ruby language in general and the Ruby on Rails framework in particular. Rails comes with a high-quality JavaScript library called Prototype, and I have written several columns describing how to use Prototype, as well as the Scriptaculous effects library that builds upon its language improvements.

Prototype isn't going away. But, over the past few months, I've noticed a growing interest—from within the Rails community and from other developer communities as well—in jQuery, another high-quality, open-source JavaScript toolkit. Most significant, jQuery was chosen by Microsoft as the official JavaScript library for its developers. jQuery also has a large number of ready-made plugins, including many that provide user-interface functionality. And, as I've started to explore jQuery, I'm beginning to think that its fans have a point, and I've even started to consider switching some of my work away from Prototype to jQuery.

What makes jQuery so special and different? What does it offer? And, how can you integrate it into your applications? This month, I try to answer all of these questions, as we explore some of the basic features of jQuery. Next month, we'll look at some of the UI widgets that jQuery provides to spruce up our sites and make them more functional.

jQuery Basics

jQuery was first released in 2006, based on preliminary work that John Resig had done since August 2005, as a simple JavaScript library that would make it more convenient to develop Web applications. Over time, it has grown to include many contributors. Resig himself has written two books on JavaScript and now works for the Mozilla corporation as a JavaScript evangelist.

If you have used Prototype previously, you won't be surprised to know that $() is not only a legitimate function name in JavaScript, but it's also used extensively within jQuery. However, $() works differently in jQuery than in Prototype. In Prototype, you can say:

```
$('foo')     // Prototype
```

to get the element with an id attribute of foo, or:

```
$('foo', 'bar')     // Prototype
```

to get the elements with id attributes of foo and bar. The number of parameters to Prototype's $() determines whether it returns a single value or an array, as well as how many elements that returned array contains.

Prototype also lets you retrieve items using CSS selectors (and a variety of pseudo-selectors), using the $$() function. For example:

```
$$('tr.even')    // Prototype
```

returns an array (and always an array, even if it matches only a single object) of all of the tr tags with a class of even.

Well, jQuery works similarly, except that it has only a single function, $(). That function is smart enough to recognize what you want, based on a single CSS-style selector that you give it. (And yes, you may specify only a single selector.) However, id attributes need to begin with a # character, as is the case in CSS. Thus, you can say:

```
$('#foo')    // jQuery
```

to get all the tags (and there should be only one such tag) that have an id attribute of foo, and:

```
$('tr.even')  // jQuery
```

to get all the tr tags with a class of even. Each call to $() might return zero, one or a number of objects matching that selector. That is, if you were to say:

```
$('#blahblah') // jQuery
```

and there isn't any such item, jQuery happily will return the set of elements matching that query—an

empty set.

This might seem a bit weird at first. After all, don't you need to know in advance how many results you'll get, or even if there will be results at all? Otherwise, how can you know whether to call a function on the one returned element or to iterate over a set of elements and call the function on each of them?

Things make much more sense once you understand that many of jQuery's functions operate using what's known as implicit iteration. That is, you can say:

```
$('p').show();
```

and jQuery will grab all the paragraphs in a document and show them. If there aren't any paragraphs, nothing happens. If only one paragraph matches, it is shown (if it wasn't showing already). The idea that you don't have to use an each() loop to go through each element is a powerful one, and it repeats itself often in jQuery code.

Equally powerful is the fact that most jQuery methods return the same set they were passed. For example, if we want to show all paragraphs and then make their background color red, we could say:

```
$('p').show().css({'background-color': '#f00'});
```

Chaining methods together in this way is quite typical in jQuery, and it helps make code more readable and concise.

When I first saw the way jQuery uses $(), I realized this meant I probably wouldn't be able to use both jQuery and Prototype in the same program, because there would be a namespace collision between $() in each of the two systems. However, the authors of jQuery thought about this very issue and have made it possible to load jQuery without activating $(), making it possible to mix jQuery and Prototype in the same file:

```
jQuery.noConflict();
```

Although I'm not sure that it's a good idea to go into a project planning to mix JavaScript libraries, there are times when you might want to use a particular effect or widget, which is available in (for example) YUI, but not jQuery.

Effects and AJAX

Like many other JavaScript libraries, jQuery comes with a set of visual effects that you can use on a page. We already have seen mention of show and hide, although each of these also can take an argument (slow, normal or fast, or an integer) indicating the speed at which the effect should run:

```
$('p').show('slow').css({'background-color': '#f00'});
```

Similarly, you can have elements hide and reveal themselves by sliding (slideUp and slideDown) or fading (FadeIn and FadeOut). And, of course, you always can modify one or more CSS attributes, as we saw in an earlier example, overriding their original settings.

Event Handlers and AJAX

It's easy to set an event handler in jQuery. For example, if you want to pop up an alert every time someone clicks on the button with an id attribute of mybutton, you would write:

```
$('#mybutton').bind('click', function() {
        alert("Hello, there!");
    });
```

Because it is so common to bind an event handler to the click of a button, you can shorten it to:

```
$('#mybutton').click(function() {
        alert("Hello, there!");
    });
```

The thing is, where do we put this event handler? We can't put it in the <head> of the document, because the <body> (in which the button presumably is contained) is not yet defined and available. We could assign it to a DOM handler, but there are issues associated with that. The jQuery method is both unobtrusive (as modern JavaScript should be), effective and cross-browser-compatible. We register an event handler for when the document is ready, and then put any event handlers we want in there:

```
$(document).ready(function() {
    $('#mybutton').click(function() {
            alert("Hello, there!");
        });
}
```

If you put this in the <head> of your HTML file, jQuery will execute the function when the document is ready. This means by the time the HTML is rendered for the user, event handlers all will be in place. It is not uncommon for the invocation of `$(document).ready()` to contain the key JavaScript invocation code for a site, with long, complex functions placed in an external library.

Like other JavaScript libraries, jQuery also provides built-in support for AJAX (behind-the-scenes HTTP) requests. For example, we can make it such that clicking on a button sends an HTTP request to a server, grabs the returned HTML snippet and then

puts that snippet in the user's browser window. To do that, we need an HTML file:

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="screen"
     href="test.css"/>
    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
      // JavaScript code here
      $(document).ready(function() {
          $('#mybutton').click(function() {
              $('#message-paragraph').load('blah.html');
          });
      });
    </script>
  </head>
  <body>
    <h1>Test page</h1>

    <p id="message-paragraph">This is a paragraph</p>
    <p><input type="button" id="mybutton" value="Button" /></p>
  </body>
</html>
```

In the head of the file, we have a standard call to $(document).ready, which assigns a handler to the click event on the button at the bottom of the page, whose id attribute is mybutton. The function, very simply, tells the paragraph (message-paragraph) to load the file blah.html from the same origin (that is, server) as the current page. The browser retrieves the file in the background, asynchronously, allowing the user to do other things while the contents of blah.html are retrieved and then stuck into the appropriate paragraph.

The above demonstrated that jQuery can retrieve HTML from an external file. But, jQuery can do more than that, retrieving not only HTML, but also XML and JSON (JavaScript Object Notation) over the network and then parsing it. We even can load a JavaScript file and execute it:

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="screen"
     href="test.css"/>
    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
      // JavaScript code here
      $(document).ready(function() {
          $('#mybutton').click(function() {
              $.getScript('blah.js');
          });
      });
    </script>
  </head>
  <body>
    <h1>Test page</h1>

    <p id="message-paragraph">This is a paragraph</p>
    <p><input type="button" id="mybutton" value="Button" /></p>
  </body>
</html>
```

Then, I create blah.js:

```
$('#message-paragraph').html("<h1>Boo!</h1>");
```

In other words, I've split the functionality from one file into two different ones. The difference is that the loaded file is treated as a program and is executed as such. So, when I click on the button, the contents of blah.js are loaded by jQuery, which then modifies the paragraph.

Conclusion

As I hope you've seen, JavaScript programming with jQuery is fairly easy and straightforward, and it allows us to do many things quickly and elegantly. Next month, we'll continue to look at jQuery, examining its plugin architecture and some of the widgets in jQuery's UI library.■

Reuven M. Lerner, a longtime Web/database developer and consultant, is a PhD candidate in learning sciences at Northwestern University, studying on-line learning communities. He recently returned (with his wife and three children) to their home in Modi'in, Israel, after four years in the Chicago area.

Resources

jQuery is distributed from the Web site **www.jquery.org**. That site not only has software downloads, but also documentation, tutorials and links to various libraries and tools that many jQuery authors use.

I recently received two books on jQuery, both of which I found to be quite good. From Packt Press comes *Learning jQuery* by Jonathan Chaffer and Karl Sweebber, which is good for Web developers who have experience in another language already—perhaps even JavaScript. It reviews many of the different types of functionality a JavaScript programmer can accomplish using jQuery.

A new book that aims more for the basics is *JavaScript: The Missing Manual* by David Sawyer McFarland, published by Pogue Press and O'Reilly Media. This is a good book for JavaScript beginners, and it uses jQuery for many of its examples, particularly in the second half of the book.

EmperorLinux

...where Linux & laptops converge

Powerful Linux: The Rhino

- Based on the Dell Precision M6400/Latitude E6500
- High performance NVidia 3-D on a WUXGA widescreen
- High performance Core 2 Quad, 16 GB RAM
- Ultimate configurability — choose your laptop's features

Features include:

- 2.2-3.0 GHz Core 2 Duo/Extreme (dual-core) or 2.5 GHz Core 2 Quad QX9300 (quad-core)
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3700M graphics, 128 core CUDA
- 80-320 GB hard drive (7200 rpm SATA) or 128 GB solid state drive, up to 16 GB RAM
- DVD±RW or Blu-ray, Ethernet, 802.11a/g/n, Bluetooth
- One year Linux tech support - phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:

Portable Linux

Toucan T500/W500

- ThinkPad T500/W500 by Lenovo
- Up to 15.4" WUXGA w/ X@1920x1200
- ATI Radeon graphics
- 2.2-2.8 GHz Core 2 Duo
- Up to 4 GB RAM
- 100-320 GB hard drive / 128 GB SSD
- 5.3-6.5 pounds
- Starts at $1280

Tablet Linux

Raven X200 Tablet

- ThinkPad X200 tablet by Lenovo
- 12.1" WXGA w/ X@1280x800
- 1.2-1.86 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-320 GB hard drive / 128 GB SSD
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at $2065

Rugged Linux

Tarantula CF-30

- Panasonic Toughbook CF-30
- Fully rugged MIL-SPEC-810F tested: drops, dust, moisture, & more
- 13.3" XGA TouchScreen
- 1.6 GHz Core 2 Duo
- Up to 4 GB RAM
- 80-320 GB hard drive
- Call for quote

www.EmperorLinux.com 1-888-651-6686

Web Site Creation Tools You've Never Heard Of

MARCEL GAGNÉ

In a world of Web 2.0 applications, AJAX applications and content management systems with countless plugins, you might think the humble Web site is a thing of the past. Not true. But, when working with these sites, we turn to the same tools over and over again. Time for something a little unusual, *non?*

I thought you said this Web site was basically a single page, François, and yet you've been at it for hours. Are you having trouble tuning in to those creative waves? Oh, I see, you haven't even started the page. Still trying to set up a PHP content management system, looking for plugins and trying out themes? Seems like a lot of work for a one-page site. That house your Aunt Guylaine is trying to sell will have sold itself before you get her page up on our site. Now, now, François, I'm not trying to be mean. I'm simply suggesting that you might be working a little too hard, and it is getting late. Our guests are already starting to arrive, *mon ami*. Look sharp.

Good evening, everyone, and welcome to *Chez Marcel*, where fine wine is paired with delectable open-source software. Please, sit and make yourselves comfortable. Tonight's wine is a Sella & Mosca Cannonau Sardegna Riserva 2005, a rather intense Italian red that certainly will capture your attention. François, please head to the wine cellar and bring up a couple cases.

When it comes to HTML and site creation tools, there's comfort in the familiar. KDE users create with Quanta Plus, and GNOME users code with Bluefish. Yet, plenty of other tools exist; some you may never have heard of. Tonight, I'd like to introduce you to a few. Of course, you don't always need to *create* a Web page. All you need is a Web version of something you already have. For instance, simple tools are available designed specifically to convert one or another document format into HTML. Suppose you wanted to show off your rather sweet Perl script in HTML format, with syntax highlighting. That might take some pretty tedious HTML coding in an editor. There is an easier way.

The faithful re-creation of code can be tough, especially with all those angle brackets, ampersands and other special characters that permeate many languages. For that, we have code2html, a rather clever little program that takes your code and turns it into great-looking HTML (Figure 1).

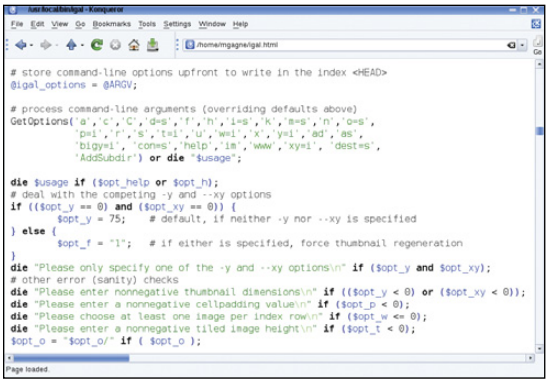


Figure 1. A Perl script is converted to clean, easy-to-read HTML, courtesy of code2html. Extra points if you can identify the script.

The basic form of the command is:

```
code2html your_code > somefile.html
```

It certainly is possible that code2html won't be able to figure out what kind of code you are giving it. These are called modes in the program's notes, and you can display all the modes by calling `code2html -m`. Let's assume that the program couldn't make out a particular shell script:

```
code2html -o html-dark -l shellscript
➥/etc/rc.d/rc.local > ~/rc.html
```

Of course, I did add a couple additional flags. From the -m output, I found that the mode for a shell script was shellscript, which I passed using the -l flag. The -o flag tells the program to produce a dark background HTML page as opposed to the default white.

There are similar programs to generate HTML code from a variety of sources. Some operate locally to generate static pages, and others can act as CGI scripts and produce dynamic text (like `man2html`,

for example).

What about the humble Microsoft Word, doomed forever to live in a proprietary format? Sure, you could find people with a copy of Microsoft Office and have them save the document as HTML, but why go through all that trouble? One very useful program I've used in the past is called wv. More accurately, Dom Lachowicz's wv is more of a collection of tools, including a library for creating filters within other programs. Some of these programs convert Word documents (2000, 97, 95 and others) into PDF (wvPDF), plain text (wvtext) and, yes, HTML (wvHtml), to name just a few. The real plus of using something like wvHtml is that you can batch-convert a whole collection of documents via a shell script.

To convert your .doc format file to HTML, use the following command:

```
wvHtml filename.doc newfile.html
```

If there are embedded images, they will be extracted with links added to the HTML file.

Eventually, however, you may need to do a little HTML coding yourself. Although it's not difficult to learn, basic HTML does require you to follow that particular language's syntax as you mark up your document for presentation. Even if you do know HTML, most people don't want to type out every tag and attribute manually. That's why we have HTML editors—to make that tedious work somewhat less tedious. In keeping with my theme of obscure, largely unknown Web creation tools, allow me to introduce a few HTML editors you likely have never heard of.

The first is HTMLpage, a simple HTML editor written in Python—and, I do mean simple. Regular visitors to this restaurant will know that I occasionally cover things for reasons that include fun as well as education. Given that this editor is basically a Python program, with plain-text code easily viewed and edited, it's also a great little program for learning and tweaking a little Python. Nevertheless, this oh-so-simple editor has some handy features, such as automatic table generation and conversion of links as well as basic text to HTML. There's a color widget for selecting and inserting color codes. The editor even supports drag and drop of page elements, such as graphics, directly into the editing window—all this in a few hundred lines of Python code.

To use HTMLpage, simply extract the package into a folder of your choosing and execute the HTMLpage.py file from there. An editing window appears with the opening and closing HTML tags automatically inserted. From there, you can enter your text in between the body tags. Some things are pretty cool for such a simple program. For instance, enter your text, select it, and click the HTML-ify button. Paragraph and line breaks are taken care of automatically. Select a link (Figure 2), click that same button and the proper tags are inserted.

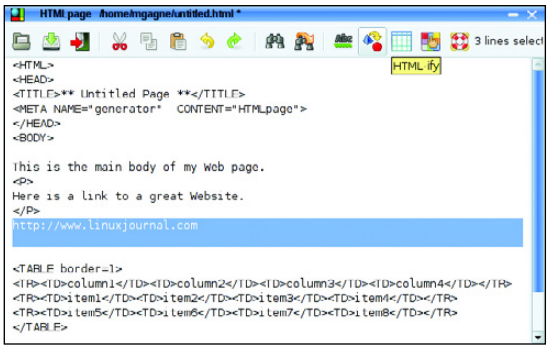


Figure 2. HTMLpage, though modest in design, has some interesting automatic features, like HTML-ify and Table-ify.

Want to create a table? Simply enter your text separated by tabs. On the second (and third and fourth) line, do the same until you have all your data. Select it, and then click the Table-ify button. Your information is inserted into a table automatically. When you click the HTML-ify button, just as when you click the Table-ify button, a little magic takes place beneath the surface. The result looks like what is shown in Figure 3.

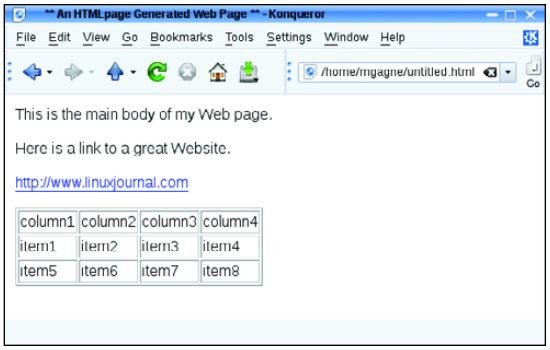


Figure 3. In no time flat, the page elements come together.

Of course, if you are going to create something of any complexity, you will need something a little more interesting than HTMLpage. So, let's look at the second HTML editor you've probably never heard of.

While my faithful waiter refills your glasses, perhaps what you really need for that simple Web site or page is a spot of TEA.

Peter Semiletov's TEA is a light, but full-featured HTML editor written in Qt (Figure 4). It's small, fast and contains a surprising number of features,

including some you won't find in the larger, shinier Web site design tools. Aside from the obvious HTML tag edits, TEA has a tabbed layout, template and scripting support (Python, Perl, Ruby and so on), Bookmarks, syntax highlighting, drag and drop into the editing window (such as for images), Wikipedia editing and a whole lot more. There's even a Morse code translator—seriously.

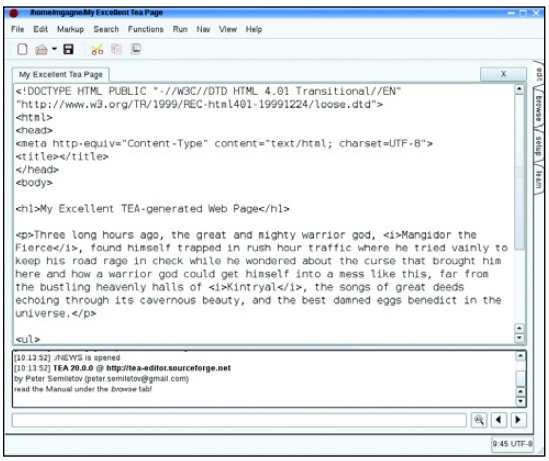


Figure 4. The TEA HTML editor is available in both GTK and Qt. Active development, however, is Qt-only.

To try out the latest-and-greatest TEA, you'll likely have to build it from source (although some packages are available on the site), but that's relatively easy. Extract the source into a folder of your choice, and type `qmake` from inside that folder, followed by `make install`. The result is a single executable called tea. A slightly older version of TEA exists called teagtk (Peter recently shifted development from GTK to Qt). Although it's not as up to date, it should be in many repositories, just waiting to be downloaded.

Along the right-hand side of TEA's editing window, you'll see four tabs. These give you access to the editing window (which is itself multitabbed, one for each open document), TEA's built-in file browser, the setup page and, finally, a manual labeled learn. Below the main window, there's a scrolling status window that displays your most recent action. The file browser makes it easy to insert bookmarks anywhere inside your directory tree for quick access. TEA also has a special quick-access file called Crapbook (accessible under the File menu) into which you can scribble quick notes regarding your current project.

The best place to start, after clicking the New button, is in the function menu. While you can create and store templates to get you started on a project or page, clicking Function, then Place from the menu bar lets you insert a basic HTML template to start your document (Figure 5). Notice as well that all the menus are tear-off—there's a dashed line above each one. Simply click the dashed line and drag the menu to your desktop. You might want to do this with the Markup menu for basic HTML tagging.

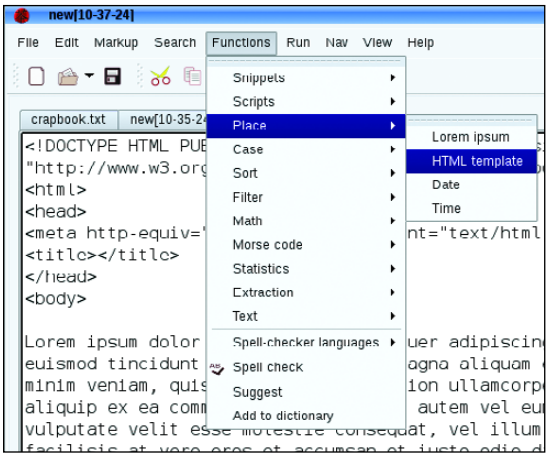


Figure 5. TEA's menus are all tear-off. Simply click on the dashed line at the top of a menu and drag it to your desktop. And, yes, that says Morse code.

Let's move on to another editor you've probably never heard of. Screem, written by David Knight, is a great site creation tool written for the GNOME desktop environment (Figure 6). You can, of course, run it under KDE, as I am doing at this moment. If the name seems a bit frightening, cast your worries aside. Screem is an acronym for Site CReation and Editing Environment, and in that respect, it is aptly named. Some of Screem's features include a number of wizards to insert special characters (entities), generate a form, create a table or provide you with an easy way to select color.

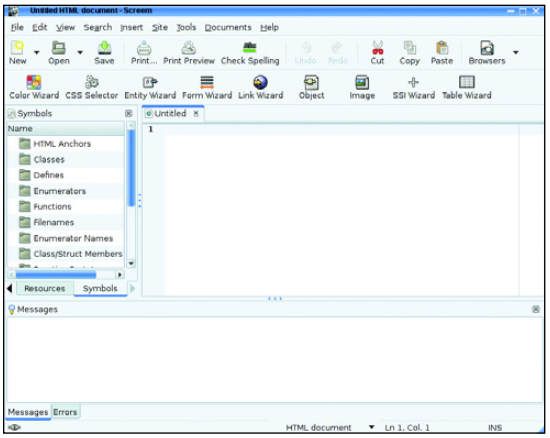


Figure 6. The Screem HTML editor and site creator is nothing to be afraid of.

NOTE:

Like the others I've covered so far, Screem is also a code editor, as opposed to a WYSIWYG (what you see is what you get) editor. Strangely, there are advantages to a code editor that may not be immediately apparent if you like the idea of building a Web page the way you would create a word-processing document. The big advantage is control, although there certainly are others. A code editor lets you see exactly what your HTML tags look like, making it possible for you to produce exactly the code that you want. The HTML produced in code editors occasionally tends to be a bit cleaner as well.

Before you get going on that first Web site, I am going to direct you to the Preferences dialog for a quick change of one of the default settings. Click Edit, then select Preferences. A six-tabbed window appears from which you can modify the default

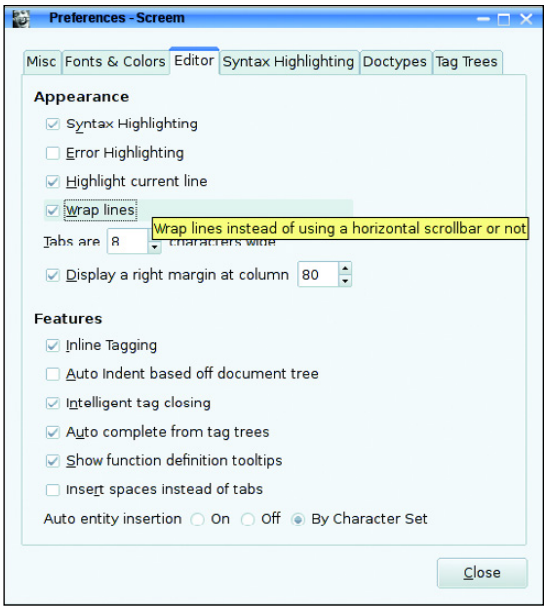


Figure 7. I'm guessing you'll want to make sure word wrap is turned on in the editor's Preferences dialog.

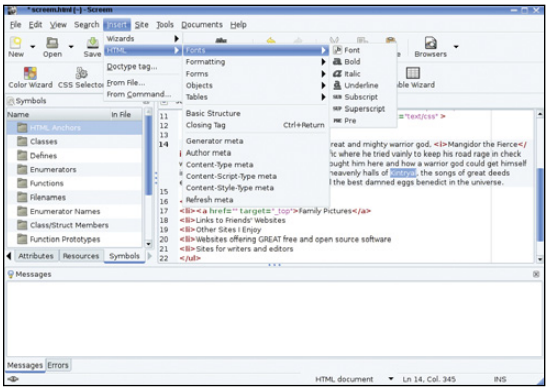


Figure 8. The vast majority of Screem's functions and markup are found in the menus. The toolbars are reserved for core tools and Wizards.

operation of much of the Screem's interface and features. There is one setting that I recommend you change right away and that's word wrap. Click the Editor tab and look for the check box labeled Wrap lines (Figure 7). Make sure it's checked on, and then click the Close button.

You can, if you want, take some time here to familiarize yourself with the other settings. Before I move on, however, let me direct you to one more that may interest you. Under the Misc tab, there is a timed backup feature (set at ten minutes by default) that you might want to activate. My personal favorite keyboard combination is Ctrl-S (used in most editors), and I tend to press it every couple minutes whether I need to or not. The autosave feature can take care of that habit.

The edit window, where you type your text, is that big empty space on the left. When editing a page (Figure 8), you highlight text, click Insert, and then select your HTML markup from the submenu. Other markup elements, such as a link to another Web site, are best done using the wizards. These are on the second toolbar, but also under the same Insert menu.

On the right, there is a multitabbed sidebar. The tabs access a built-in file manager, a tag tree from which you can jump to any tag in any part of the document quickly, an attributes view that further defines all tags and their attributes, and more.

As you may have noticed, *mes amis*, the clock is telling us that closing time is upon us. While my faithful waiter refills your glasses a final time, remember this. The free and open-source software landscape is extremely rich with countless projects and programs available for your downloading pleasure. The easy path is certainly the one that installs the most popular programs, such as Quanta, the KDE HTML editor, or Bluefish, the GNOME favorite. Yet, there are many other projects, and as with a bottle of wine, it can be fun and educational to try those you've never encountered before. You even may discover a new favorite. Please, *mes amis*, raise your glasses and let us all drink to one another's health. *A votre santé! Bon appétit!*■

Marcel Gagné is an award-winning writer living in Waterloo, Ontario. He is the author of the *Moving to Linux* series of books from Addison-Wesley. Marcel is also a pilot, a past Top-40 disc jockey, writes science fiction and fantasy, and folds a mean Origami T-Rex. He can be reached via e-mail at marcel@marcelgagne.com. You can discover lots of other things (including great Wine links) from his Web sites at www.marcelgagne.com and www.cookingwithlinux.com.

Resources

HTMLpage: **www.pcbypaul.com/software/htmlpage.html**

Screem: **www.screem.org**

TEA HTML Editor: **tea-editor.sourceforge.net**

WvWare: **wvware.sourceforge.net**

Marcel's Web Site: **www.marcelgagne.com**

Cooking with Linux: **www.cookingwithlinux.com**

TECH TIP Access the X Window System Clipboard from the Command Line with xclip

Ever selected text from your terminal so you could paste it into an X application? Drop the mouse and use xclip instead. Using xclip, you simply can pipe the contents that you want to clip directly into xclip:

```
$ lspci | xclip
```

Then, go to your X application and paste the captured output into the application. xclip also lets you "paste" selected text into the terminal. Just use the -o switch to output the highlighted text from the active selection:

```
$ xclip -o
```

xclip can be found at **sourceforge.net/projects/xclip**.

—ERIK FALOR

OUTRAGED
by the high cost of Fibre Channel or iSCSI Storage ?

Switch to AoE !

SR2461

Coraid Offers a Complete Line of Clustered Modular Storage Products:

- High Performance EtherDrive® SATA+RAID Storage Appliances with 1 GigE or 10 GigE Connections
- Clustered VirtualStorage™ Appliances *(a Revolutionary Logical Volume Manager)*
- Scalable NAS Gateways *(File Sharing with EtherDrive® Storage)*

Coraid's EtherDrive® Storage with AoE is fast, reliable disk storage that's easy to use. And it's much more affordable than iSCSI or Fibre Channel!

- Coraid products use open AoE (ATA-over-Ethernet) block storage protocol, for high performance without the TCP/IP overhead
- With AoE, your shared storage capacity is infinitely scalable – at a fraction of the cost of iSCSI or Fibre Channel storage
- We provide a 3-year warranty and free firmware upgrades on all our products, as well as support from first-rate engineers trained in our technology

Product Excellence Award Winner LinuxWorld Conference & Expo Best Storage Solution

EtherDrive® Storage has a field-proven track record and is 1000+ large data storage customers strong.

CORAID

The Linux Storage People

To learn more about this and Coraid's other products, go online or call

+1.706.548.7200 (toll free: **877.548.7200**)

www.coraid.com

DAVE TAYLOR

Special Variables II

Using different forms of variable expansion.

Last month, we took a strange turn and actually just focused on the basics of shell scripting, special variable notation, rather than solving some complex and obscure scripting challenge. I'm going to continue our discussion this month by looking at what you can reference with a variable name. As a quick refresher, last month we looked at $0, $$, $!, $*, $? and $@.

Naming Variables

Unlike the bad-old days of coding, a few dozen extra bytes in your script have no ill effect and aren't going to eat up precious disk space, so I am a strong proponent of longer, mnemonic, descriptive variable names. Don't use i but loopcount, for example, if you want to have a variable help you step through a loop. It makes everything far easier to deal with when you go back to the script weeks or months later.

If you're already a script programmer, you know that variables are referenced by using $ + variable name. So, let's stick with account as our variable name, so I can show you some neat things.

First off, you're used to referencing the variable as $account, but you also can use ${account}, and often you have to use the full form to ensure that there are no parsing errors.

Tip: Parsing error? What if you want to output the value of account immediately followed by the digits 001? `echo $account001` will fail because the shell will think that you mean variable `account001`, which doesn't exist. Instead, use the ${} notation:

```
echo ${account}001
```

What happens if account isn't defined? When it's not defined, `echo $account`, produces a blank value. Instead, it'd be nice to say, "if the value is defined, show it; otherwise, show an alternative value." That's done like this:

```
${account:-alternative value}
```

Notice that the alternative value can have spaces embedded—another reason why the {} notation is such a winner!

How about having the same action, but also setting the variable to the specified value? That is, in longhand, the script snippet would look like this:

```
if [ "$account" = "" ] ; then
   account="alternative value"
else
   echo $account
fi
```

But, there's a delightfully short alternative: simply reference the variable:

```
echo ${account:=alternative value}
```

Maybe you want to produce an error message if the variable doesn't have a value instead? Another tiny notational change and you've got it, by George:

```
echo ${account:?No account specified}
```

One more in this punctuation soup: the `${xx:-yy}` notation displays `yy` if `$xx` isn't set or is null, but it doesn't change the value of the variable itself. I showed that a few paragraphs above. But, what if you want the opposite effect, having an alternative value shown if the variable is set? You can use:

```
${account:+alternative value}
```

Again, it won't change the actual value of the variable.

Slicing and Dicing

For this next set of cool variable name tricks, let's jump into a little demo script:

```
#!/bin/sh

account="taylor"
echo "account set to ${account:-oops, forgot to set a value}"
echo "skip the first two letters: ${account:3}"
echo "show me just the third and fifth letter:" \
     "${account:3:1} and ${account:5:1}"
exit 0
```

As you can see from this example, you can access the value of a variable from the nth letter through the end with the `${x:n}` notation. To get a specific length slice, add a third variable, `${x:n:m}`, which means "show me m letters from the variable x starting at letter n."

When I run the above script, here's what I see:

```
$ sh test.sh
account set to taylor
```

```
skip the first two letters: lor
show me just the third and fifth letter: l and r
```

Nice and simple!

Advanced Variable Tweaking

Now, let's say you have a complicated script that creates a series of variables in the form $account1, $account2, $account3 and so on. Is there a way to access all of the variable names at once? You betcha! Let's set a few variables:

```
account="taylor"
account2="smith"
account3="jones"
account4="harry"
```

Now, here's how you can access all their names:

```
${!account*}
```

It looks like this:

```
echo "variables starting with 'account': ${!account*}"
```

And, when run:

```
variables starting with 'account': account account2 account3 account4
```

To access their values, you'd just do this expansion in a loop:

```
for varname in ${!account*}
do
    echo \$varname = ${!varname}
done
```

This is a tricky situation, actually, because all of the notational conventions you might consider by default (like $$varname or ${$varname}) will fail. Instead, ${!varname} does an additional dereference step and gets what we want:

```
$account = taylor
$account2 = smith
$account3 = jones
$account4 = harry
```

I'm going to stop here, but next month we'll go further into the mysterious world of shell variable expansion and talk about built-in text substitution too.

Acknowledgement

Thanks to Scott Jangro for his help with the funky ${!x} notation example.■

Dave Taylor is a 26-year veteran of UNIX, creator of The Elm Mail System, and most recently author of both the best-selling *Wicked Cool Shell Scripts* and *Teach Yourself Unix in 24 Hours*, among his 16 technical books. His main Web site is at www.intuitive.com, and he also offers up tech support at AskDaveTaylor.com. You also can follow Dave on Twitter through twitter.com/DaveTaylor.

Samba Security, Part IV

Creating restricted shares on your secure Samba file server.

MICK BAUER

For the past four months in this column, we've been building a secured Samba server for our local LAN, using Swat. To spare those of you who have been following this series a fourth summary of the usage scenario, let's suffice it to say, we're creating a series of file shares with varying user permissions.

This month, I wrap up the series by showing how to create a restricted, "owner-only" share and how to use mount.cifs to make persistent Samba mounts on your client systems.

What We've Done So Far

Last month, we created a public share, SUPPER, and a nonpublic and group-readable share called CHORES. Prior to that, we had set up some global variables that are inherited by all shares. Those global variables were:

```
workgroup            = FED-CENTRAL
security             = user
client schannel      = yes
server schannel      = yes
map to guest         = Bad User
guest account        = nobody
unix password sync   = yes
valid users          = mick, knute, pepe, skippy, nobody
read list            = knute, pepe, skippy
write list           = mick
```

Attentive readers of Part II of this series [December 2008] may notice that I omitted "admin users" here, even though in Part II, I had set that to mick. This was an embarrassing mistake. On Ubuntu systems at least, this wreaks havoc with how Samba interacts with Linux file permissions.

You'll recall that setting "admin users" causes listed users to be logged on to Samba as root after successfully authenticating as themselves. In other words, if "admin users" is set to mick, any time mick successfully logs on to any share, he'll actually be logged on as root. The *expected* result is that mick, therefore, will have superuser privileges and won't be restricted from doing anything at all. In practice, the results tend to be much less predictable than that.

For example, on my Ubuntu 8.04 system, suppose I set "admin users" to mick, create a directory on the underlying Linux filesystem that's owned by mick and has permissions of -rwx------ (or 0700), and then I create a Samba share mapped to that directory that has no guest access or read access (that is, a share like the one I'm about to show you how to set up).

If I then try to connect to this share with this command:

```
bash-$ smbclient //CASA_DE_MICK/BUZZ-OFF -U mick
```

and enter the correct password when prompted, sure enough, silently and behind the scenes, I'll actually be logged on as root. But the result of this login will be:

```
Domain=[CASA_DE_MICK] OS=[Unix] Server=[Samba 3.0.28a]
tree connect failed: NT_STATUS_ACCESS_DENIED
```

What? How can this be? Access shouldn't be denied to anything, for root, should it? But denied it will be, if the share in question maps to a directory not owned by root. This may or may not happen on non-Ubuntu systems. My point is that using the "admin users" parameter may result in unpredictable interactions between Samba and Linux filesystems.

As I said last month, letting people use Samba shares with root privileges is dangerous anyhow. Samba client software isn't the correct tool for Samba system administration, Swat is. So now we have two good reasons always to leave "admin users" empty!

Now, let's move on to our share-specific settings. The smb.conf variables that configured SUPPER, as set via the Swat tool, looked like this:

```
path           = /home/mick/supper
read only      = yes
guest ok       = yes
invalid user   = root
valid users    =
read list      =
write list     = mick
admin users    =
hosts allow    = 192.168.44
hosts deny     = ALL
create mask    = 0644
browseable     = yes
available      = yes
```

These variables are set nearly the same for CHORES, except:

```
path            = /home/mick/chores
guest ok        = no
valid users     = +users
```

What do all these variables mean? I explained them in gory detail in the last three issues of *LJ*, and definitive descriptions can be found for all in the smb.conf(5) man page. Some of these will come into play this month too, as we create that restricted share.

Creating a Restricted Share

You'll recall that our Samba server has four user accounts: pepe, skippy, knute and mick, which correspond to my three roommates and me. These are UNIX user accounts on my Samba server's underlying OS, with corresponding but separate entries in the Samba server's separate user database. (I explained how to create and synchronize Samba user accounts in Part II of this series, in the December 2008 issue.)

For our restricted share, BUZZ-OFF, only mick should have read access *or* write access. No other user should have any rights at all on this share. Accordingly, when we create the directory to which this share will point, we'll be sure it's owned by mick and has a permissions mask of 0700 (u+rwx,g-rwx,o-rwx), like this:

```
drwx------ 2 mick users 4096 2008-11-04 00:00 buzz-off
```

Figure 1 shows the first round of parameters we'll set upon creating this share in Swat's Basic View.

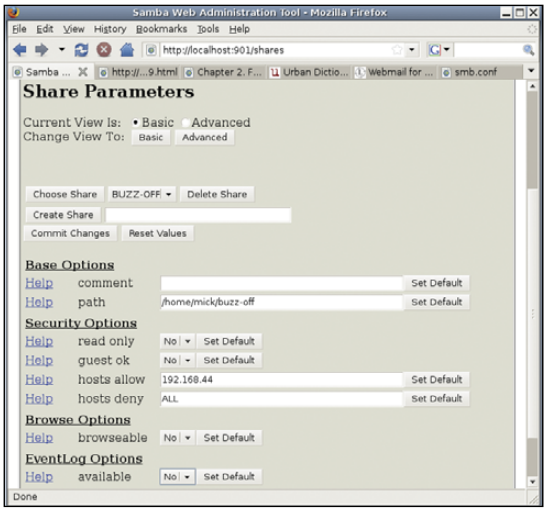


Figure 1. Restricted Share, Basic Settings

After setting the path, we set read only to no—I'll be creating new files in this share—and guest ok to no as well, because we don't want to allow any anonymous access. We'll set hosts allow and hosts deny the same as our other shares—to permit access only from the local LAN (your network address is, obviously, probably different).

We'll set browseable to no, so this share won't turn up in people's Network Neighborhood or in smbtree listings. To connect to this share, therefore, we'll need to specify its path when mapping it to a drive or connecting to it with smbclient.

And, we'll leave available set at no until we've clicked the Commit Changes button, clicked the Change View to Advanced button and changed some things in the Advanced View (Figure 2).

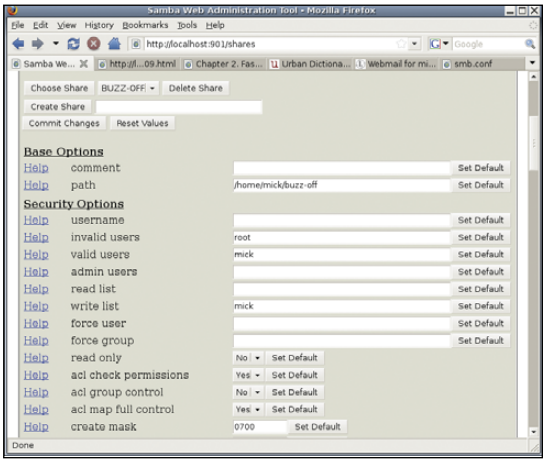


Figure 2. Restricted Share, Advanced Settings

As you can see, we're going to blank out the list of valid users except for mick and completely empty the contents of read list. The write list, however, correctly contains the single value of mick.

The only other setting we need to change is create mask, which we'll set to 0600. This is different from the 0700 mask we used when creating the directory itself; the directory's execute bit needs to be set so the directory can be used, but the *contents* of this directory, which is what the share represents, do not.

Now we can change available to yes and click the Commit Changes button. Our restricted share is ready for use!

To test this, let's first make sure the share doesn't turn up in the local Samba browse list. We can perform this test using smbtree, like so:

```
bash-$ smbtree -N -b
FED-CENTRAL
    \\CASA_DE_MICK    iwazaru-ubuntu server (Samba, Ubuntu)
        \\CASA_DE_MICK\print$    Printer Drivers
```

```
\\CASA_DE_MICK\SUPPER    Mick's menus
\\CASA_DE_MICK\CHORES    Chore lists
\\CASA_DE_MICK\IPC$      IPC Service (iwazaru-ubuntu server
                         ➥(Samba, Ubuntu))
```

Sure enough, the new share BUZZ-OFF doesn't appear in the browse list. But, is it nonetheless usable by mick, its owner? Let's find out with smbclient:

```
bash-$ smbclient //CASA_DE_MICK/BUZZ-OFF -U mick
Password: ********
Domain=[CASA_DE_MICK] OS=[Unix] Server=[Samba 3.0.28a]
smb: \>
```

It worked. I've got a Samba prompt! There's no reason not to try a quick directory listing before exiting:

```
smb: \> dir
  .                           D        0  Tue Nov  4 23:17:34 2008
  ..                          D        0  Tue Nov  4 23:17:34 2008
  access-log_10312008.txt           665  Tue Nov  4 23:17:34 2008

        52008 blocks of size 262144. 13229 blocks available
smb: \> exit
```

Everything worked as expected. One last test—just to be sure, I want to try logging in to the share as a guest user. Remember that our Samba server is set up to treat any login involving a nonexistent user name as a guest login:

```
bash-$ smbclient //CASA_DE_MICK/BUZZ-OFF -U totallyfakeuser
Password: ********
Domain=[CASA_DE_MICK] OS=[Unix] Server=[Samba 3.0.28a]
tree connect failed: NT_STATUS_ACCESS_DENIED
```

It failed, in the expected and appropriate way. Our restricted share is accessible, insofar as we want it to be.

Using mount.cifs for Persistent Samba Mounts

Now that I've got a restricted share available to me, suppose it will contain things I need to read and change on a regular basis. Do I need to access it via an interactive smbclient shell every time?

Of course not. The ability to mount remote Samba shares as though they were local volumes is one of the best things about Samba. You can do this by using the standard mount command, along with Samba's mount.cifs module, on your Samba client systems.

On Red Hat-derived and SUSE systems, the cifs filesystem and associated utilities are included with the standard samba-client package. On Debian, Ubuntu and other Debian derivatives, however, you'll need to install the package smbfs.

The ability to mount remote Samba shares as though they were local volumes is one of the best things about Samba.

Although based on the same protocols, smbfs and cifs are actually two different things. cifs is newer than smbfs, and the smbmount command formerly used for mounting Samba file shares via the smbfs filesystem has been deprecated by the Samba team in favor of cifs and the mount.cifs module. smbfs and smbmount are still distributed with Samba, but they are not being actively maintained.

While we're installing Ubuntu packages, you'll also want the package winbind, which mount.cifs needs in order to resolve NetBIOS or Windows NT names (the Samba server we've been setting up uses NetBIOS name resolution, not Windows NT). SUSE users will need the package samba-winbind. I'm not positive, but I believe winbind is included in Red Hat/CentOS/Fedora's samba-client package.

After installing winbind, you should add the string `wins` to the `hosts:` line in /etc/nfsswitch.conf (only root can do this; you'll need to use su or sudo).

After mount.cifs and winbind are in place, you're ready to start mounting Samba shares. To to this manually from a command line, you can invoke the mount command as root or, as shown here, using sudo:

```
myclientlaptop-$ sudo mount -t cifs //CASA_DE_MICK/BUZZ-OFF
 ➥./mymountpoint -o rw,suid,user=mick
```

In this example, we're telling mount to use a filesystem type (-t) of cifs. We're mounting, obviously, the share BUZZ-OFF on the server //CASA_DE_MICK, using the mountpoint ./mymountpoint (which is an existing directory within my current working directory). Note that I can, if necessary, use my Samba server's IP address rather than its NetBIOS (or Windows NT) name, in which case, that part of the command would look something like `//192.168.44.123/BUZZ-OFF`.

The `-o` gives a list of options for this mount. The first option, `rw`, lets me both read files from and write them to the share. `suid` causes any set-uid bits on files in the share to be acknowledged. `user` passes my Samba user name to the mount.cifs module so it can authenticate the session. After entering the above command, I'll be prompted first for the root password and then for mick's password.

Whatever you do, do not enter your password

on the command line using the `password=` option. Because shell commands may be logged in various places and are stored in shell histories, it's generally a terrible idea to use any password as a command argument.

If your Samba credentials are unimportant, for example, because they do not correspond to any user account with actual shell access, it's still a good idea to avoid passing its password to a command. A better option in that scenario is to use a credentials file, which is simply a text file containing a user name and password.

However, that method is *not* appropriate for storing any credentials you actually use to log in to systems. Even with strict file permissions set, it may be possible for some unauthorized person or process to copy or read the credentials file.

As with any type of filesystem mounting, you can save yourself typing by creating an entry for your mount in /etc/fstab. For the example we just used, a corresponding fstab entry would look like this:

```
//CASA_DE_MICK/BUZZ-OFF /home/mick/mymountpoint cifs
➥rw,noauto,suid,user=mick 0 0
```

As you can see, this line is very similar to the mount command line we used earlier. One new option here is `noauto`, which causes this line to be ignored at system startup—this Samba share won't be mounted until you issue a mount command, like this:

```
myclientlaptop-$ sudo mount /home/mick/mymountpoint
```

sudo will prompt you for the root password. (Again, if you aren't running Ubuntu, you could omit the sudo command and instead execute the rest of the command from a root shell session.) Then, mount will prompt you for mick's password.

Assuming authentication succeeds, you'll be able to use BUZZ-OFF as if it were part of your local filesystem, located in /home/mick/mymountpoint. When you're done working, you can unmount it like this:

```
bash-$ sudo umount /home/mick/mymountpoint
```

If you prefer your Samba mount to be activated every time your system starts, you can omit the noauto option in your fstab entry. However, unless you use a credentials file, you'll need to be present during each startup in order to enter the Samba password when prompted; otherwise, your startup will wait for you indefinitely. On a laptop system this probably isn't a problem, but on other types of systems it very well could be an issue.

Similarly, if your Samba server is unavailable for some reason when your client system starts up, this also can cause the client startup to hang or delay. When in doubt, stick to noauto mounting.

Conclusion

And, that's it for this series on Samba security. Funny how four columns can add up to only a basic tutorial, but I hope you've found it useful. Until next time, be safe!■

Mick Bauer (darth.elmo@wiremonkeys.org) is Network Security Architect for one of the US's largest banks. He is the author of the O'Reilly book *Linux Server Security*, 2nd edition (formerly called *Building Secure Servers With Linux*), an occasional presenter at information security conferences and composer of the "Network Engineering Polka".

Resources

"Samba Security, Part I", *LJ*, November 2008:
www.linuxjournal.com/article/10224

"Samba Security, Part II", *LJ*, December 2008:
www.linuxjournal.com/article/10256

"Samba Security, Part III", *LJ*, January 2009:
www.linuxjournal.com/article/10292

TECH TIP Use SSH to Create an HTTP Proxy

SOCKS is built in to OpenSSH, so it's a trivial matter to set up a local SOCKS proxy with the -D flag. For example:

```
$ ssh -D 12345 myuser@remote_ssh_server
```

will open up the port 12345 on your local machine as a SOCKS proxy so all your HTTP traffic can be specified to go through the SSH tunnel and out remote_ssh_server on the other end. Your proxy server is now set up.

Next, set up your browser to use the proxy server. Most browsers include proxy support. For Firefox 3, go to Edit→Preferences→Advanced→Network→Settings, and specify that you want to use a Manual Proxy, localhost, port 12345, and SOCKS v5 (although OpenSSH supports both versions 4 and 5).

Now your browser is using a secure tunnel to your remote SSH server.

—RICH LUNDEEN

KYLE RANKIN

Chopping Logs

Why wait for your awstats job to finish when you need custom log results now? Check out a quick-and-dirty Perl one-liner that creates speedy tallies from log files and is easy to tweak to suit your particular statistics needs.

If you are a sysadmin, logs can be both a bane and a boon to your existence. On a bad day, a misbehaved program could dump gigabytes of errors into its log file, fill up the disk and light up your pager like a Christmas tree. On a good day, logs show you every clue you need to track down any of a hundred strange system problems. Now, if you manage any Web servers, logs provide even more valuable information in terms of statistics. How many visitors did you get to your main index page today? What spider is hammering your site right now?

Many excellent log-analysis tools exist. Some provide really nifty real-time visualizations of Web traffic, and others run every night and generate manager-friendly reports for you to browse. All of these programs are great, and I suggest you use them, but sometimes you need specific statistics and you need them now. For these on-the-fly statistics, I've developed a common template for a shell one-liner that chops through logs like Paul Bunyan.

What I've found is that although the specific type of information I need might change a little, for the most part, the algorithm remains mostly the same. For any log file, each line contains some bit of unique information I need. Then, I need to run through the log file, identify that information and keep a running tally that increments each time I see the particular pattern. Finally, I need to output that information along with its final tally and sort based on the tally.

There are many ways you can do this type of log parsing. Old-school command-line junkies might prefer a nice sed and awk approach. The whippersnappers out there might pick a nicely formatted Python script. There's nothing at all wrong with those approaches, but I suppose I fall into the middle-child scripting category—I prefer Perl for this kind of text hacking. Maybe it's the power of Perl regular expressions, or maybe it's how easy it is to use Perl hashes, or maybe it's just what I'm most comfortable with, but I just seem to be able to hack out this kind of script much faster in Perl.

Before I give a sample script though, here's a more specific algorithm. The script parses through each line of input and uses a regular expression to match a particular column or other pattern of data on the line. It then uses that pattern as a key in a hash table and increments the value of that key. When it's done accepting input, the script iterates through each key in the hash and outputs the tally for that key and the key itself.

For the test case, I use a general-purpose problem you can try yourself, as long as you have an Apache Web server. I want to find out how many unique IP addresses visited one of my sites on November 1, 2008, and the top ten IPs in terms of hits.

Here's a sample entry from the log (the IP has been changed to protect the innocent):

```
123.123.12.34 - - [01/Nov/2008:19:34:02 -0700] "GET
 ➥/talks/pxe/ui/default/iepngfix.htc HTTP/1.1"
 ➥404 308 "-" "Mozilla/4.0 (compatible; MSIE 7.0;
 ➥Windows NT 6.0; SLCC1; .NET CLR 2.0.50727;
 ➥Media Center PC 5.0; .NET CLR 3.0.04506; InfoPath.2)"
```

And, here's the one-liner that can parse the file and provide sorted output:

```
perl -e 'while(<>){ if( m|(^\d+\.\d+\.\d+\.\d+).*?
➥01/Nov/2008| ){ $v{$1}++; } } foreach( keys
 ➥%v ){ print "$v{$_}\t$_\n"; }'
 ➥/var/log/apache/access.log | sort -n
```

When you run this command, you should see output something like the following only with more lines and IPs that aren't fake:

```
33      99.99.99.99
94      111.111.111.111
138     15.15.15.15
```

For those of you who know and love both Perl and regular expressions, that one-liner probably isn't too difficult to parse, but for the rest of you, let's go step by step. Sometimes it's easier to go through a one-liner if you see it in a formatted way, so here's the Perl part of the one-liner translated as though it were in a regular file:

```
#!/usr/bin/perl
```

```
while(<>){
    if(m|(^\d+\.\d+\.\d+\.\d+).*?01/Nov/2008|){
        $v{$1}++;
    }
}

foreach( keys %v ){
    print "$v{$_}\t$_\n";
}
```

First, let's discuss the while loop. Basically, `while(<>)` iterates over every line of input it receives either through a pipe or as a file argument on the command line. Inside this loop, I set up a regular expression to match and pull out an IP address. The regular expression is probably worth looking at in more detail:

```
(^\d+\.\d+\.\d+\.\d+)
```

This section of the regular expression matches the beginning of the line (^), then any amount of numbers (\d+), and then a dot, another series of numbers, another dot, another series of numbers, another dot and finally a fourth series of numbers. This pattern will match, for instance, 123.123.12.34 at the beginning of a line. I surrounded this part of the regular expression in parentheses. Because this is the first set of parentheses, when Perl matches it, it puts the resultant match into the $1 variable so I can pull it out later.

Now, those of you who know regular expressions know that I cheated here. This regular expression is not very explicit at all. For one, it would match completely invalid IP addresses, such as 999.999.999.999. For another, it even would match any series of four numbers with dots in between, such as 12345.6.7.8910. I chose an overly generic regular expression on purpose to make a point. There are explicit regular expressions that match only valid IP addresses, but those expressions are very long, very complex and, in this case, completely unnecessary.

Because I'm dealing with Apache logs, I am pretty confident that the first set of numbers at the beginning of the file is an IP address and not something else, and second, the IP address that Apache logged should be reasonably valid. In taking the shortcut, I not only saved on typing, but the resulting regular expression also is easier to read and understand even if you aren't a regex wizard.

After I match the IP, I want to match only log entries from November 01, 2008:

```
.*?01/Nov/2008
```

This section performs matches on any number of characters (.*), and with the question mark at the end, it matches only as much as it needs to and no more. Then, it matches the datestamp for November 01, 2008. If I wanted a tally of every day in the log file, I could omit this entire section of the regular expression. Alternatively, if I wanted to match on some other keyword (for instance, when the user performed a GET on a particular file), I could replace or augment the above section with that keyword.

Once I have matched the IP address in a line and have assigned it to $1, I then use it as a key in a hash I call %v here and increment it (`$h{$1}++`). The power of a hash is that it forces each key to be unique. That means each time I come across a new IP, it will get its own key in the hash and have its value incremented. So, if it's the first time I see the IP, its value will be one. The second time I see the IP, it will increment it to two and so on.

Once I'm done iterating through each line in the file, I then drop to a foreach loop:

```
foreach( keys %v ){
    print "$v{$_}\t$_\n";
}
```

Basically, all this does is increment through every key in the hash and output its value (the number of times I matched that IP in the file) and the IP itself. Note that I didn't sort the values here. I very well could have—Perl has powerful methods to sort output—but to make the code simpler and more flexible, I opted to pipe the output to the command-line sort command. That way, even if you don't know Perl too well but know the command line, you could tweak arguments in sort to reverse the output or even pipe it further to tail, so you could see only the top ten IPs.

If I want to know only the overall number of unique visitors, as each line represents a unique visitor, I just need to count the overall number of lines. To do this, I simply need to pipe the output to `wc -l`.

And, there you have it, a quick-and-dirty one-liner to chop through your logs and tally results. The beauty of using Perl hashes for this is that you can tweak the regular expression to match all sorts of values in the file—not just IP addresses—and tally all sorts of useful information. I've used modified versions of the script to count how many times a particular file was downloaded by unique IPs, and I've even used it to perform statistics on mailq output.■

Kyle Rankin is a Senior Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *Knoppix Hacks* and *Ubuntu Hacks* for O'Reilly Media. He is currently the president of the North Bay Linux Users' Group.

Polywell Linux Solutions

More Choices, Excellent Service, Great Value!

Serving the Industry for More Than 20 Years

Netdisk 8000V
Quiet Performance NAS Storage

The new generation of energy-efficient 45W AMD Athlon™ X2 dual-core processors with Cool'n'Quiet™ technology are designed to enable quiet performance.

4TB $1,399
8TB $2,399
12TB $2,999

- Dual Gigabit LAN
- RAID-5, 0, 1, 10
- Hot Swap, Hot Spare
- Linux, Windows, Mac
- E-mail Notification
- Tower or Rackmount

AMD Athlon™ X2 64

Silent Eco Green PC

Based on the new 45W AMD® Athlon™ X2 dual-core processor is energy efficient, quiet and has plenty of power. starts at **$199**

Fanless Silent ITX PC

1G DDR2, 80GB Hard Drive starts at **$299**
AMD® processor, Low-profile Add-on Available

AMD Opteron™ 64

4U 24Bay Storage Server

AMD Opteron™ Processors
36TB RAID-6 128GB RAM, 4 x GigaLAN

AMD Phenom™ 64

Small 1U Server for Data Center ISP

AMD Phenom™ processor
4 to 8GB DDR2-800, 2 x 500GB RAID HD
Linux Server Starts at **$499**

Polywell OEM Services, Your Virtual Manufacturer
Prototype Development with Linux/FreeBSD Support
Small Scale to Mass Production Manufacturing
Fulfillment, Shipping and RMA Repairs

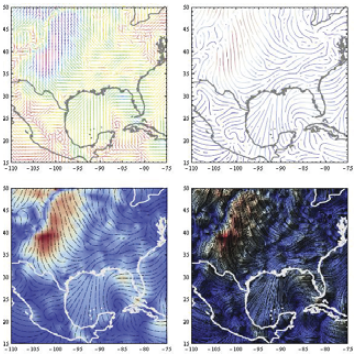
- 20 Years of Customer Satisfaction
- 5-Year Warranty, Industry's Longest
- First Class Customer Service

888.765.9686
linuxsales@polywell.com
www.polywell.com/us/Lx

Polywell Computers, Inc 1461 San Mateo Ave. South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

POLY POLYWELL

Wolfram Research's Mathematica



After rocketing its Mathematica application from version 6 to 7 in just 18 months, Wolfram Research's developers may need testing for blood-caffeine content. Mathematica is a powerful general computation environment for calculations, large-scale computations, complex programming, and visualizing and modeling data. After dubbing Mathematica 6 the "most important advance in its 20-year history", Wolfram says that version 7 is a major release that adds 500 new functions and 12 new application areas. Added functionality includes image processing, built-in parallel HPC, new on-demand curated data and other new computational innovations. The firm further claims that this release has made parallel computing mainstream. Mathematica 7 has 32- and 64-bit editions for Linux x86, Solaris UltraSPARC/x86, Windows and Mac OS X.

www.wolfram.com

Neuros Technology's Neuros LINK and Neuros.TV Service

The Neuros folks definitely think like we do. Their new Neuros LINK is a hackable, nonproprietary set-top box that connects the television to the Internet via existing open Internet standards. Neuros positions the device "squarely between the dedicated, proprietary electronics devices and the powerful but clunky and expensive personal computer". One can play downloaded content in virtually any format from any source. Neuros also claims to have created a navigation structure that makes the LINK experience "feel like TV browsing rather than Web browsing". Currently, the LINK is a Gamma Product—that is, the post-beta, white box preproduction stage especially geared for hackers and hard-core early adopters. Meanwhile, the accompanying Neuros.TV is a free service that enables Neuros LINK users to find, organize and share Web-based video content.

open.neurostechnology.com

Fixstars' Yellow Dog Linux

Same dog, new tricks, different owner. In other words, the popular Yellow Dog Linux (YDL) for the Cell Processor has been upgraded to version 6.1 and is now under the tutelage of the company Fixstars. The Tokyo-based Fixstars recently acquired Terra Soft Solutions, the company long associated with YDL. Fixstars states that YDL 6.1 is unique in that it drives both the desktop and development environments forward simultaneously. For end users, YDL v6.1 offers an improved graphical wireless configuration tool and the ability to use PS3 video RAM for temporary storage or swap. For developers, it offers advancements such as the new Cell Superscalar and support for the Cell's programming model. Supported platforms include Apple G4/G5, Sony PS3, YDL PowerStation and IBM Power Systems.

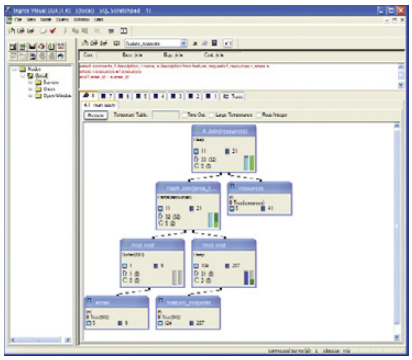
Yellow Dog Linux™

us.fixstars.com

NICTA and Infocomm Research's Maritime Wireless Mesh Network

Though still in the prototype stage, a wireless mesh network for ships has been announced by two research institutions: Australia's NICTA and Singapore's Institute for Infocomm Research. Under normal conditions, the maritime system provides data and voice communications between port authorities, container terminals and ships via shore-based WiMax inter-ship connectivity. In poor weather conditions, the system utilizes a backup satellite system. Because, says NICTA, standard VoIP and other data-transport techniques don't work well with satellite systems, the system utilizes onboard mesh nodes and NICTA's mobile routers to handle the satellite connectivity. The shore-based system aims to deliver a 6Mbps long-range (20km) ship-to-ship and ship-to-shore mesh communication system capable of ad hoc multi-hop communication with other vessels and shore command stations.

www.i2r.a-star.edu.sg, www.nicta.com.au

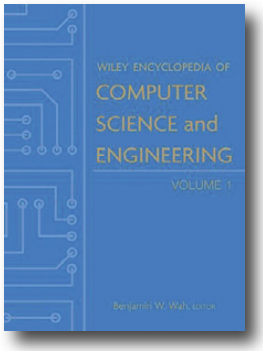


Ingres Corp.'s Ingres Database

If we open-source enthusiasts have anything to brag about, it is a great selection of robust databases. A fine case in point is Ingres Corporation's Ingres Database, which was just upgraded to version 9.2. The company bills Ingres as "flexible, simple, secure, reliable and scalable [and able to] cope with even the most complex, multilanguage requirements including business intelligence, content management, data warehousing, enterprise resource planning (ERP) and logistics management". Core advancements in version 9.2 relate to improved application development, enhanced availability and supportability, as well as the simplification and automation of many tasks traditionally associated with maintaining a business-class database.

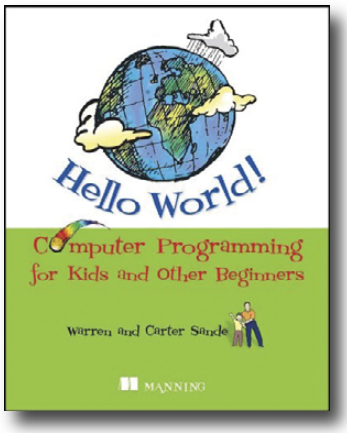
www.ingres.com

Wiley Encyclopedia of Computer Science and Engineering



While the Web provides information in bulk at your fingertips, there remains no substitute for concise, authoritative reference works that are more than a stranger's brain dump. Such is the role of Benjamin W. Wah's new *Wiley Encyclopedia of Science and Engineering*, a five-volume, 3,300-page set with more than 450 A-to-Z articles on the latest advances and findings in computer science and engineering. Some broad topics include standards, electronic commerce, financial engineering and computer education. Each article is written by experts in their particular specialty and is peer-reviewed by two others to ensure reliability.

www.wiley.com



Warren D. Sande and Carter Sande's *Hello World!* (Manning)

The father-son team of Warren D. Sande and Carter Sande think that anyone can program a computer, even a 12-year-old. The duo's new book *Hello World! Computer Programming for Kids and Other Beginners* from Manning is a "gentle but thorough introduction to the world of computer programming". Written in a manner free of "geek speak", *Hello World!* contains lots of pictures, cartoons and fun examples to hold the reader's interest. The free Python is the programming language utilized in the book. Programming concepts that are covered include memory, looping, decisions, input and output, data structures, graphics and others, which are then applied to interesting topics like computer graphics, game programming and simulations. The publisher says that *Hello World!* can be used in either a home or classroom setting.

www.manning.com

AMAX's ServMax Personal Supercomputer

In support of the needs of scientific computing, AMAX has released its new ServMax Personal Supercomputer (PSC) workstation, which it dubs "a cluster in a box". The ServMax PSC supports up to 720 processing cores and 3 Teraflops in a single workstation. AMAX asserts that the product delivers "up to 15x cost savings and 15x lower power consumption than traditional 1U rack-optimized servers". Targeted applications include life sciences, geosciences, engineering and sciences, molecular biology, medical diagnostics, EDA, government/defense, visualization and financial modeling. Other features include parallel architecture and NVIDIA CUDA technology.

www.amax.com

Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, 1752 NW Market Street, #200, Seattle, WA 98107. Submissions are edited for length and content.

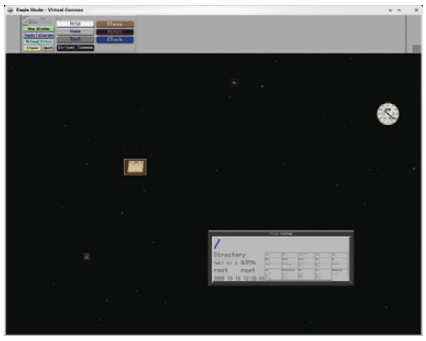
Fresh from the Labs

Eagle Mode—File Manager with a Difference

(eaglemode.sourceforge.net)

For those sick of file manager after file manager that essentially do the same thing, with only a slightly different interface, let's just say that Eagle Mode (EM) takes a more ambitious approach. In the words of EM's documentation:

> Eagle Mode is an advanced solution for a futuristic style of man-machine communication, in which the user can visit almost everything simply by zooming in. It has a professional file manager, file viewers and players for most of the common file types, a chess game, a 3-D mines game, a multi-function clock and some fractal fun, all integrated in a virtual cosmos. By featuring a separate pop-up-zoomed control view, help texts in the things they are describing, editable bookmarks, multiple input methods, fast anti-aliased graphics, a virtually unlimited depth of panel tree, and by its portable C++ API, Eagle Mode aims to be a cutting edge of zoomable user interfaces.



Eagle Mode's file manager lives within a virtual cosmos where other programs and trinkets float around with it.

For those chasing a lightweight file manager, you're looking at the wrong project. I almost choked when I read the system requirements: CPU 3.4GHz...per core! Plus, 1GB of RAM and a gig of hard drive space for temporary files at runtime—this is an ambitious project that shoots at the opposite end of the scale! Nevertheless, don't despair if you don't have all the hardware requirements; I still found the project usable on my apparently measly 512MB of RAM and 2.14GHz per core.

Installation Thankfully, the software requirements aren't as Draconian as the hardware requirements. The only major dependencies are Perl, GCC, libx11-dev and the libxine-dev library for playing multimedia files. There is a list of other smaller non-essential dependencies that is long enough to make you scroll down the page (such as tar, xterm, JPEG libraries and so on), but they're really the kind of things you would expect for a file manager (and unless you have a system that's more sparse than a nightclub in Salt Lake City, you probably have them all installed anyway).

So, head to the Web site, grab the latest tarball, extract it and open a terminal in the new folder's main directory. From here, I really recommend reading the documentation included in the doc directory, which, unlike a lot of documentation, actually is well set out and easy to navigate. But, for the impatient, enter the following commands:

```
$ perl make.pl build
```

And, as root or sudo:

```
# perl make.pl install
```

I got an error in the middle of compiling EM, which said:

```
Building emAv failed, but that project is not so essential.
So if you don't know how to solve the problem, you could
continue the overall building now, and live without the features
the project provides. Continue? [y(es)/n(o)/a(lways)]: y
```

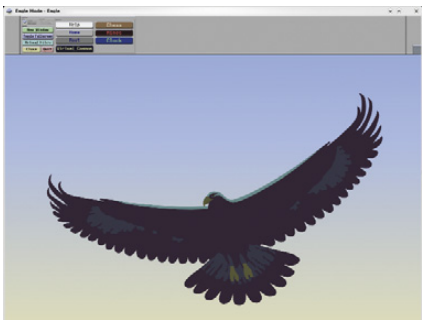
After doing some Googling, I still couldn't find out what emAv was, but it seems to be non-essential, and the installation continued on unfazed. EM seems to run fine without it. Once the compilation has finished, change into the installation directory, which by default will be:

```
$ cd /usr/local/eaglemode/
```

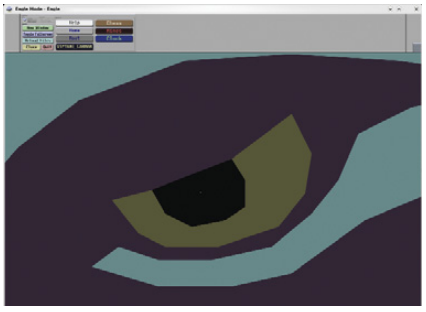
And, enter this command to run it:

```
$ ./eaglemode.sh
```

Usage Once inside EM's main screen, you'll be presented with a scene that is deceptively conventional—a file manager in your home directory. Double-click on a file, and it opens it. But, double-click on a folder, and it opens a terminal within that directory—that's weird. There's a bunch of extra info in the bottom right of the folder too, what's that about? As soon as I roll the mouse wheel upward, the whole scene suddenly zooms in, and I can see the contents of those files in detail. That's kind of cool, but it still isn't what I'd call groundbreaking. Then, I give the mouse wheel a few quick rolls backward, and the whole idea of Eagle Mode unfolds in an instant. When you zoom out, you find that everything is placed within a 3-D



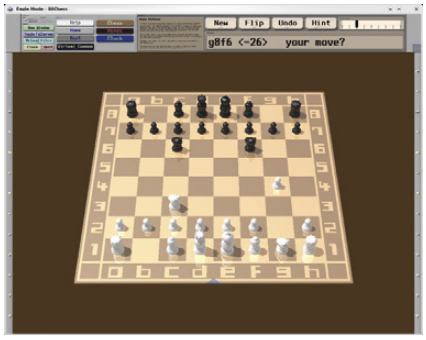
Scroll backward all the way, and it is soon revealed that your virtual cosmos is living within the eye of this proud eagle!



I'm not sure if it will come out in print, but those who look very carefully will be able to see this as a white speck in its eye. Cosmic!

virtual cosmos—a star field consisting of files, applications and fields of stars. Keep zooming out, and eventually it is revealed that you are looking at *a cosmos that is living within the eye of an eagle*—just the kind of artistic silliness of which I approve! Zooming out to the end of the universe displays this proud eagle in its entirety.

Zoom back in to the eye of the eagle and you return to the virtual cosmos where you can zoom in to programs and folders in great detail. Clicking on an area of the universe changes the "aim" of the zoom, and if you hold the scroll wheel in and move the mouse in any direction, it will scroll around the screen that way. Included in these applications and objects are things like system folders, a clock, documentation and games, and I'm sure there are some hidden objects in that star field somewhere. For those looking for a distraction, a version of *Chess* has been included (with a rather aggressive AI, it must be said) and a game simply called *Mines* that is a mind-bending 3-D take on the classic *Mine Sweeper*.



Waste time in Eagle Mode playing a rather lovely 3-D *Chess* game against a rather nasty AI.

Overall, if you're sick of navigating your PC with something that feels like it was designed by a bank manager (or if you simply want to put all those GHz your new PC came with to use), Eagle Mode just might be for you (not recommended for boring people though).

Mp3Wrap—MP3 Merger

(mp3wrap.sourceforge.net)

With podcasting becoming ever more popular, people are dealing with large groups of MP3s that have to be squished into one big file and sent out to the general public. This in itself isn't very hard, but these files generally are all re-encoded and placed in something like a run-of-the-mill 128kbps MP3. When something that already has been under lossy file compression, like an MP3, gets encoded a second time, it loses a great deal of audio quality, and the resulting sound is more like a warbly old vinyl record being pumped through a Commodore 64. Well, worry no more radio DJs; this project may be just for you. According to the Mp3Wrap Web site:

> Mp3Wrap is a free, independent alternative to AlbumWrap. It's a command-line utility that wraps quickly two or more MP3 files in one single large playable MP3, without losing filenames and ID3 information (and without need of decoding/encoding). It also provides the possibility of including other non-MP3 files, such as playlists, info files, cover images, inside the MP3. This means you obtain a large MP3 that you can split at any moment just using mp3splt, and in a few seconds, you have all the original files again! It's useful because files created with Mp3Wrap are easy to download. In fact, you don't need to know each song name to download, and it's easy to play. Even if you don't have mp3splt to split the file, you can listen to it anyway.

Installation Installing Mp3Wrap is a doddle, with a choice of a source tarball or .rpm and .deb packages. Plus, compiling the source is easy and painless. Click the DOWNLOAD link at the bottom of the home page for a list of all the package options. If you're going with the source version, grab the tarball, save it locally and extract it. Open a terminal in the new folder, and do your run-of-the-mill:

```
$ ./configure
$ make
```

And, as root or sudo:

```
# make install
```

Usage Mp3Wrap currently is a command-line-driven affair, but don't let that put you off, as it's quite simple. The syntax is as follows:

```
mp3wrap finaloutcomefile filetoadd1 filetoadd2
```

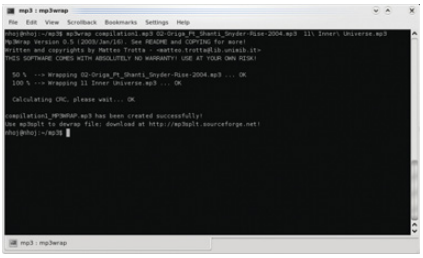
It looked like this after I had given it some files:

```
$ mp3wrap compilation1.mp3
 ➥02-Origa_Ft_Shanti_Snyder-Rise-2004.mp3 11\
 ➥Inner\ Universe.mp3
```

The terminal output does give some useful information as to what's happening. As a side note, remember that for some reason the program will insert _MP3WRAP just before the .mp3 extension as a sort of identifying mark, so if you're doing something like writing a shell script and having trouble finding your compilation MP3 file, that will be why.

A bunch of useful switches are included, the best of which is -a, allowing you to add more MP3 files to an existing "wrapped" MP3. Another useful switch is -l, which when passed a wrapped MP3 will list whatever files are inside. Check the man page for more details.

A drawback of the command-line nature of Mp3Wrap is that it may become very tiring and strenuous when dealing with a long list of MP3 files (which probably will lead to some mistakes with long playlists). Also, although Mp3Wrap's files are usable on just about anything that will play MP3s, they do have trouble seeking



The mix-tape is back! Mp3Wrap lets you compile multiple MP3s into one large file *without* losing any sound quality.

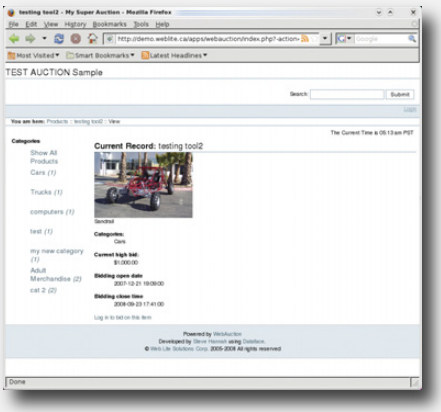
in some older players, such as XMMS and the like. This project is just begging for a GUI front end (which its cousin application mp3splt already has), as a GUI on top would make things much easier for a radio DJ on Friday night and would avoid the likely mistakes that will come from compiling a playlist of songs via command-line switches. Teething problems aside though, this program is a very clever one that will give podcasters a distinct edge over their rivals with original rip quality in their songs, and it might find its way into the hearts of many MySpace emo types looking to make an awful "mix-tape" MP3 compilation for some budding emo on-line girlfriend. Radio DJs and sad teenagers rejoice!■

John Knight is a 24-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

Project at a Glance

Web Auction

(**apps.weblite.ca/webauction**)



Web Auction

Sick of selling on eBay and getting slugged by seller fees? Or, would you simply like more control by having your own Web auction? Well, Web Auction (imaginative title, I know) might be exactly what you need. Designed for organizations or individuals, Web Auction is simple and quick to use, and it differs from eBay in that only administrators can add products, giving you full control over your auction. You can host an auction by yourself, but the Web Auction folks currently are letting you host auctions on their servers for free, which is jolly nice of them!

Brewing something fresh, innovative or mind-bending? Send e-mail to knight.john.a@gmail.com.

HARDWARE

The Incredible Shrinking Laptop

A review of the Dell Mini 9.

DANIEL BARTHOLOMEW

I think of the range of different laptops in the world as falling into three basic models: mini, standard and huge. The huge laptops are those with 17" (or larger) screens. The standard laptops are those with 14"–15" screens. Mini laptops are a new breed. Popularly known as Netbooks, they come with small 9"–10" screens, small keyboards, 802.11b/g/n wireless adapters and usually no optical drive. For me, the huge laptops are much too large to bother with. I consider standard-size laptops the perfect trade-off between portability and functionality—they're big enough to have full-size keyboards, and the screens are decent in size, but they still are small enough to be fairly portable. There are, of course, many laptops that fall between or outside these three categories, but they provide a good starting point for me.

Figure 1. The size difference between the Mini and the D610 is striking.

Figure 2. The D610 looks like it could swallow the Mini and have room for dessert.

Standard and huge laptops have been around for a while, and most of the recent excitement in the laptop world has centered around the mini or Netbook segment. I was curious whether one of these ultraportable Netbooks would make a compelling replacement for the old Dell Latitude D610 I've been carrying around for several years, so I picked up the recently released Mini 9. The Mini is Dell's entrant into the Netbook market.

The Hardware

True to form, there are several options from which to choose when purchasing a Mini 9 from Dell. These include all the standards like a larger hard drive, more memory, integrated Bluetooth and a built-in Webcam. You even can choose to "upgrade" the Ubuntu 8.04.1 OS that the base model comes with to Windows XP. Why anyone would want to do that is beyond me, but the option is there if you want it.

I chose to keep things simple and get the base model. I did this for a few reasons, the first of which was the nice $349 ($373 after taxes) price tag. The second reason is so many reviews cover fully loaded machines with every option possible, which I think leads to a false sense of capability. For this review, I wanted to explore exactly how good the base model is.

The base Mini 9 (at the time of this writing) comes with an Intel Atom processor N270 running at 1.6GHz and a 533MHz 512K L2 Cache. It also comes with 512MB of RAM, a 4GB SSD (solid-state drive), 8.9" screen, 802.11g wireless

networking and a 32-Watt-hour four-cell battery. Unfortunately, no Webcam is included in the base model.

The build quality of the Mini is very good. The screen hinge is precise and reliable, and the case has little to no flex. It just feels solid. The ports are pretty standard: three USB ports (two on the left side, one on the right), an SD/MMC/memory stick card reader, a VGA port, headphone and microphone jacks, and an Ethernet port. With everything else on the Mini, including the keyboard, display and trackpad, being shrunk, it is nice that the ports are the standard full-size variants instead of proprietary miniature versions that require special cables you can't find anywhere.

It comes as no surprise that the D610 has the Mini beat in the ports department. It has all the ports the Mini has (except the card reader) as well as an additional USB port, a DVD drive, S-Video out, Modem, parallel port and serial port. It makes up for the lack of a built-in card reader with a PCMCIA slot, which coincidently enough, I have filled with a four-in-one card reader.

The resolution of the Dell Mini 9's 8.9" screen is 1024x600. The width of the screen is good for most Web sites. The 600-pixel height normally would not be enough in my mind, but the Mini gets around this by turning the Windows key into a dedicated "full-screen" button that works in most applications. I'm glad Dell did something with that key, as otherwise it would be a wasted space on a keyboard that's cramped enough already. I've actually caught myself pressing the windows key on my other systems when I wanted to take an application full screen. The D610 has a 14" screen, but the resolution is practically the same: 1024x768. The extra height of the D610 screen is nice, but it's not enough to give it a clear advantage over the Mini.

One additional note about the screen on the Mini is that it's very bright, easily beating the D610. The screen also is viewable in almost all lighting conditions. The D610 screen is easily overpowered in sunlight, so the Mini has a definite advantage there.

The speakers on the Mini are nothing special. They sit on either side of the Dell logo beneath the screen, and they get the job done. They're not as loud as the speakers on the D610, but the sound quality is similarly average. For everyday listening on either laptop, a good pair of headphones is the best choice.

The trackpad on the Mini is molded in as part of the case plastic instead of being a separate piece like on the D610. Dell wisely chose to keep the left and right mouse buttons below the trackpad instead of moving them off to the side like other Netbook manufacturers. The sensitivity and accuracy beats the trackpad on the D610 easily, but the finger nub on the D610 is better than either trackpad. My preference is to use a mouse whenever possible, but I can live with the Mini's trackpad when a mouse is not available.

The four-cell battery the Mini comes with has been good for 3.5–4.5 hours of battery life, depending on the load to which I have subjected it. I can't remember what the battery life was on the D610 when it was new, but the Mini beats it by at least an hour now.

The Software

Dell calls the OS installed on the Mini the "Mini OS powered by Ubuntu 8.04". The main difference between it and regular Ubuntu 8.04 is the desktop replacement software, which gives you handy shortcuts to your most-used applications instead of the normal Ubuntu desktop. It also dispenses with the bottom GNOME panel and opts instead to put everything up at the top to save space.

One of the primary things that attracted me to the Mini is its inclusion of Ubuntu pre-installed. My D610 runs Ubuntu fine, and almost everything "just works" on it. By "almost" I mean everything except the wireless driver, which tends to break every time I upgrade to a new version of Ubuntu or apply an especially big update. I can fix the wireless easily, but it's a pain to have to do so as often as I do. Because the Mini comes with Ubuntu pre-installed, my hope was that everything would "just work" out of the box with no effort on my part, and Dell did not disappoint.

Boot times are in the minute range, which is not super speedy, but also not slow enough to be annoying.

When booting the Mini the first time, Dell displays a few notices about where and how to request service, should it be required, and then leads you through the process of creating an initial user and making various setting choices, including language and login preferences.

One thing the setup does not do is prompt you for your networking settings. Instead, you are expected to configure this after you log in the first time. That wasn't a big deal to me, but I think it could lead to a "what do I do now?" moment for novice users.

As mentioned before, after logging in to the Mini 9 you are, by default, shown a custom launcher interface instead of a regular Ubuntu desktop. The launcher comes configured with a generic set of categories: Productivity, Web, Entertainment, Games and Learn. Under each one is a set of applications. The oddest choice in my mind is that the Nautilus file manager is stuck in the Entertainment category. The categorization of other applications makes more sense: OpenOffice.org Writer is under Productivity, Rhythmbox is under Entertainment, Firefox and Pidgin are in the Web group and so on. All in all, the selection of apps is nice and surprisingly broad, and new users will find plenty to keep them occupied for a good long while.

Figure 3. The Mini has an attractive desktop with several custom wallpapers.

Figure 4. The Productivity category has links to the various OpenOffice.org products and to Acrobat Reader.

The categories and apps in the groups can be modified easily. When you hover your mouse over a group or application tile, a little teardrop icon appears in the upper-right corner. Clicking on that spins the tile around and presents you with a small configuration dialog. You can change the icon that appears on the tile, the name of the tile and, in the case of launcher tiles, you can configure the application, folder or Web site the tile opens.

Apart from the launcher, the rest of the system is regular Ubuntu. The APT package management works wonderfully, and the Mini comes configured to use a special Mini-9-only repository hosted by Canonical. It contains everything I've tried to install, but it doesn't track in lockstep with the main Ubuntu repositories—for example, at the time of this writing, Ubuntu 8.10 is not available through it, but it does provide timely security updates.

With only four gigabytes to work with, disk space is an issue, but not as much as I thought it would be. As it comes from the factory, the Mini uses about 3GB of disk space for the OS and applications, leaving a single gigabyte free on the base model. Instead of loading this remaining space with media files, I put them on an SD card, and that arrangement has worked very well. With a couple 8GB or 16GB SDHC cards, I don't think I will ever lack for "space" on the Mini.

The Mini comes preconfigured with several useful add-ons, including Java, Adobe Flash and Adobe Acrobat. There also are several add-ons that are not so useful (to me anyway), such as the Yahoo Toolbar and the Dell Video Chat program. Of course, if I were a Yahoo user or had purchased the integrated Webcam, those add-ons probably would be on my useful list, so I can't knock them too much.

Dislikes

Not everything is rosy in Mini-land. There are a few things that I just do not like or cannot seem to adapt to.

The biggest complaint I have with the Mini is that the keyboard is cramped. In fairness, I expected the keyboard to be cramped, and in some ways, the keyboard is better than I expected. The alphanumeric keys are nearly full size, with the punctuation and modifier keys on the left and right sides half as wide. The keyboard has a good feel, and I would be perfectly happy with it if not for a few *big* problems I have with the keyboard layout. The first issue I have with the layout is the single (') and double (") quotes key has been moved from the home row to the bottom of the keyboard next to the arrow keys. This is a stupid place for a key that I use all the time. There's a reason this key is on the home row. My little finger ends up hitting the Return key all the time whenever I want to quote anything.

One of the primary things that attracted me to the Mini is its inclusion of Ubuntu pre-installed.

Figure 5. The Mini's keyboard is full of compromises—some good, some bad.

Another issue I have with the keyboard layout is some keys have been pushed off the regular keyboard entirely and can be accessed only while pressing the Fn key. These include the function keys (F1–F10), the braces keys ({, }, [and]), pipe (|), backslash (\), accent (`) and tilde (~) keys. I'm sure most people won't miss many of those keys, but for me, the difficulty in getting to them is an annoyance that prevents me from doing much of any shell scripting or even long-winded blog posts on it.

The last issue I have with the keyboard, and which my fingers are as yet unable to get used to, is the dash (-) and equals (=) keys have been pushed down one row to make room for the Delete and Backspace keys. Instead of being to the right of the number keys, they are to the right of the P key (where the braces keys should be). In typing terms, this means I keep pressing Delete every time I try to enter a dash or underscore.

I can't be too hard on the keyboard though, because the fact is, there's just not enough space to put every key where my fingers think it should go. What we get is a compromise, and like all compromises, there are things I like and things I don't.

Another issue I have with the Mini is time-related. When resuming from sleep, the Network Manager takes an additional 10–20 seconds or so after the computer wakes up before it gets the network up and running. This extra waiting is annoying, but there's probably not much that can be done about it.

I have also experienced an occasional Network Manager glitch where it refuses to stay connected to my wireless router and refuses to auto-connect when I log in. Usually a reboot fixes it. It may be that Network Manager and I just don't get along.

Conclusion

The Mini 9 from Dell is a great little laptop. It is very capable and can do just about everything I want a laptop to do. For those who can adapt to working with a smaller keyboard, it is at least as capable as any three- or four-year-old laptop, like my D610, without the driver headaches. And, if portability is a big concern, it is hard to beat the dimensions and weight of the Mini. It is the only laptop I've used that fits into my motorcycle tank bag, which is a big plus in my book.

However, I have decided not to keep it. I'm not sending it back or anything; instead, I'm going to give it to my eldest daughter for her birthday. She's old enough for her own computer, and her hands fit the keyboard better than mine. I've also let her "borrow" it a few times over the past few weeks, and she thinks it's "really cool" (her words). But, don't tell her that she's getting it, I want it to be a surprise.■

Daniel Bartholomew lives with his wife and children in North Carolina. His on-line home is at daniel-bartholomew.com.

Living, Reinvented:

The Technology of Abundance and Constraints

O'REILLY®

ETech™

Emerging Technology Conference

MARCH 9–12, 2009
SAN JOSE, CA

conferences.oreilly.com/etech

ETech, the O'Reilly Emerging Technology Conference, is O'Reilly Media's flagship "O'Reilly Radar" event. Since 2002, ETech has brought to light the disruptive yet important innovations that we see on the horizon, rather than the ones that have already arrived, honing in on what's going to be making a difference not this year, or maybe even next year, but around the corner as the market digests the next wave of innovation surprises.

In 2009, our focus is on Living Reinvented: The Technology of Abundance and Constraints, where we'll examine how emerging technologies can bridge the gap between the worlds of material plenty and economic constraint. We'll look at the trends, tools, and technologies that that operate at the intersection of abundance and scarcity, and how both these worlds cross-pollinate, collaborate and influence each other.

CONFERENCE TOPICS

- City Tech
- Mobile & The Web
- Health
- Objects & Materials
- Geek Life
- Wireless Signals
- GeoLocation
- Developing Markets
- Computing

FEATURED SPEAKERS AT ETECH 2009

Jane McGonigal
Avant Game

Mark Frauenfelder
Boing Boing/Make Magazine

Photo by Sara Forrest

Mary Lou Jepsen
Pixel Qi

Joichi Ito
Creative Commons

REGISTER AND SAVE 15%

When you use discount code **et09lj**

HARDWARE

The Archos 5

Sure, it can do a lot, but can it do what you want?

DANIEL BARTHOLOMEW

Figure 1. The Archos 5 comes with what you see here. The strange piece of plastic is an adapter for the DVR Station add-on.

For the past couple years, I've carried around a Cowon D2 media player. This little device has 2GB of onboard memory (there also are 4GB and 8GB versions). It also has a built-in SD slot for storage expansion. I purchased it because it plays every audio format I care about: MP3, Ogg Vorbis and FLAC. As a bonus, it also can play 320x240 MPEG 4.2 AVI video files.

This little player has served me very well, but lately I've become interested in getting a portable media player that is as good at video as the Cowon is at audio. Specifically, I'm getting tired of converting videos just so I can play them on my D2. I'd like a player that can handle unmodified versions of all of my media, and a lot of onboard storage would be a nice perk.

Cowon has some higher-end devices that look like they might make compelling replacements, but they run Windows Mobile, which I don't want. There also is the ubiquitous iPod from Apple, but I'm not really interested in getting one of those either. With Cowon and Apple out of the running, I went searching for alternatives, and the first possibility I encountered to replace my D2 was the Linux-based Archos 5.

Archos 5 Hardware

Physically, the Archos 5 is quite a bit larger than the D2. The 5 in Archos 5 refers to the screen size, which actually clocks in at 4.8 inches. Measurements aside, the screen feels more than twice as large as the 2.5" screen of the D2, partly due to the higher resolution (800x480) display, which is a good bit more than twice the D2's 320x240 resolution (at least in width).

The screen is very glossy—a trend in screens I am not very fond of—but it is quite viewable under most conditions, even though it is a little too shiny for my tastes.

The thickness of the Archos 5 varies based on which model you get. The 60GB version appears (from the images on the Archos Web site) to be thinner than the D2, and the 120GB and 250GB models appear to be about twice the thickness of the 60GB model. I picked up the 120GB version, and part of me wishes I had the thinner model, despite the smaller drive size.

Rounding out the exterior of the Archos 5, there is a power button, volume control button, a reset hole, a headphone jack, a pair of docking ports on the bottom and a sturdy foot that pivots out the back to prop it at a nice viewing angle.

In addition to the base device, you can purchase several add-ons for the Archos 5 that give it new functionality. These include a TV antenna, a DVR, a helmet-mountable video camera, an FM radio and a GPS. Although I don't foresee myself purchasing any of these add-ons, they certainly prove that the base hardware is

Figure 2. Archos 5 and Cowon D2

Figure 3. The Mini-Dock comes with a multi-national power adapter, A/V cables and a real, honest-to-goodness USB cable.

capable of a great many things if attached to the right accessories.

Archos 5 Software

The GPL'd portions of the Archos 5 source code are not available from the Archos Web site at the time of this writing. The source code hopefully will be posted by the time you read this.

Archos bills the Archos 5 and the larger Archos 7 as "Internet Media Tablets". As such, they include many features one doesn't necessarily associate with a simple media player, such as a Web browser, e-mail client and various widgets.

The Web browser on the Archos 5 is Opera-embedded. It is a little slow, but renders most pages well. AJAX-heavy sites, like Google Reader, seemed to give it the most trouble. It has Flash support, and for video sites like YouTube and Google Video, it will offer to play the video full screen, which is a nice touch. The Web browser has tabs that allow you to have multiple sites open at the same time. You also can zoom in (and out) on pages by double-tapping on the screen.

When downloading large files, the download screen takes over the interface and does not let you queue other downloads or continue browsing. That annoyance aside, it will try to put files you download into the proper folders automatically (Video, Music and so on), which is nice. If the Archos 5 doesn't recognize a file, such as zip or tar.gz files, it places the file in a Downloads folder that you can access the next time you connect the Archos 5 to your computer.

Overall, I am quite pleased with the built-in Web browser. It's not something I plan to use often, but it works well enough for light browsing when I am away from my desk.

The mail client can connect to both POP and IMAP mail servers. The client is functional, but a bit clunky, and it is not something I'm likely to use unless it is the only choice I have.

The photo viewer is nice and turns the Archos 5 into a decent digital photo frame. In the photo viewer, a vertical top-to-bottom swipe will rotate the picture clockwise, and a vertical bottom-to-top swipe will rotate the image counter-clockwise. Horizontal swipes will move to the next and previous pictures.

The audio player lets you sort music in all the standard ways: by genre, artist, year, album, title and so on. The Web radio section of the audio player is powered by vTuner and has a nice selection of stations. Local music (not streaming music from over the local network or Internet) can play behind the slideshow or Web browser.

The video player is pretty basic. You simply navigate the folders of the Archos 5 and choose the video you want to play. Once one video has finished, the next one starts, just like in the audio player.

If you have to interrupt local audio or video playback, the Archos remembers where you were, and you can resume from where you left off when you next access either mode.

Thanks to the built-in Wi-Fi and the appropriate software, the Archos 5 can connect to UPnP servers on the local network. This has turned out to be one of my favorite features. It turns the Archos 5 into a sort of roving satellite television for the house. Both my file server and my Popcorn Hour media player are set up as UPnP servers, so the Archos has access to all my media. Well, it would, if it could play all my media (more on that later).

Some pieces of software on the Archos 5 are missing. One of these is the File Sharing item in the tools menu, which reports that you need to update your firmware to gain access to the feature (even though I'm running the latest firmware).

Rounding out the software on the Archos 5 are several widgets that provide such things as a simple newsreader, a currency converter, a note-taking app and a weather widget. They're not terribly useful; the newsreader contains only seven preconfigured entries that can't be changed, for example, but they're there to play with if you want. The widgets actually run in a special mode of the Web browser, and the newsreader feeds open stories in tabs of the browser.

Things I Don't Like

For a device as hefty (price- and size-wise) as the Archos 5, it has a lot of deficiencies.

My biggest gripe is that there is no out-of-the-box support for playing h.264-encoded video. This is a major limitation. It's not a question of ability, because there's a plugin you can purchase to enable it (which I did reluctantly). The plugin bundle to enable h.264 video, AAC audio, HD video support and MPEG-2 video support costs 30 euros, which worked out to about $43 at the time I purchased it. The three plugins separately are 15 euros each, so I guess I'm getting a good deal, but it just feels like Archos is trying to fleece me. On top of that, the HD plugin is not yet available.

I can understand paying extra for physical hardware that provides the Archos with new abilities. The helmet cam, the GPS and the DVR station are good examples of this. But to cripple the main unit out of the box deliberately by not playing h.264 video seems greedy on Archos' part—especially considering that the base unit costs $350–$450 (depending on the model).

A related gripe is that the Archos 5 will play only video that is smaller than the size of the screen (800x480). As a test, I cropped the 480p version of *Big Buck Bunny* so that it was exactly 800x480 pixels in size, and the Archos refused to play it. The largest files I have been able to play are in the neighborhood of 720x460. The HD plugin supposedly will allow the Archos 5 to play up to 720p-size video when it is finally released.

Another video-related issue is that the Archos 5 has trouble with h.264 .m4v files that are longer than 1.5 hours. It will report that the file is corrupted, even when it is not. Lengthy non-h.264 .avi files do not have this issue.

Moving on to audio, the Archos 5 is frankly disappointing. There are no technical reasons that I can see why the Archos 5 cannot play FLAC and Ogg files, but it can't. They don't even show up in the list of files. The Archos also had trouble with my example .m4a audio files. They could be viewed in the music browser, but they would not play. AAC audio plays fine when part of an .mp4 video file, so it can play it.

The Archos comes up short in regard to media playback, but I knew some of that going in, thanks to the specs on the Archos Web site. What was not anticipated were the number of hardware issues I had, and they bear mentioning here. At the top of the list is that the Archos 5 comes with a proprietary USB cable that is used for connecting the Archos 5 to your computer and for charging.

This cable is inadequate for several reasons. First, it takes eight hours to

charge the Archos 5 via this USB connection. Battery life for the Archos 5 in my testing was about three hours for video and about double that for audio, so the included charging "solution" can't even keep up with normal use! This contrasts with the 50+ hours of audio playback I regularly get out of the Cowon D2 in between chargings.

The long charge times for the Archos also mean that once I watch about 1.5 movies, I am done for the day. Second, the cable is not a standard USB cable. One end is USB; the other is a proprietary docking connector. I don't mind docking connectors, but when that is the only connection to the outside world, I get nervous and annoyed.

My first hardware accessory purchase for the Archos 5 was the $30 mini-dock. Although it looks strange when plugged in to the Archos 5, at least it provides a real power adapter and recharge times of less than three hours. The mini-dock also gives the Archos 5 component A/V and S-Video out, as well as USB host and client support. The mini-dock adds so much functionality and utility to the Archos 5, I wish the company had incorporated the ports on it into the main unit.

On the software front, some of the widgets—the newsreader and weather widgets in particular—need a network connection to run, but if the network is turned off, they will give you a "No network available" message, instead of starting up the network like the browser and mail applications do. To use them, you first must activate the network manually and then run them. Compounding the frustration of this is that the Archos 5 attempts to save power by shutting off the network automatically if it hasn't been used in a few minutes.

I also ran into some out-and-out crashes when using the Archos 5. I can't count the number of times the Archos has rebooted itself during what I consider to be normal use. It happens at least once a day, often more.

The first time I powered up the Archos 5 and connected to my home network, it said there was a firmware update available. That was all well and good, and it downloaded and applied the update like I expected, except that after the update, the wireless settings were erased, and then the Archos decided that it also could not see my, or any other, access point. Not cool. A reboot fixed it, but the update could have indicated that an additional reboot was required on top of the one it did during the update process.

The biggest error I ran into was when trying to copy a bunch of pictures over to the Archos 5, it froze, and I had to force unmount the device and reboot it. Then, when mounting the device, on multiple computers, it would mount only read-only. I ran fsck on the drive, like so:

```
sudo fsck.vfat -a /dev/sdf1
```

and then I was able to mount and write to the Archos 5 again. It also recovered the pictures that had been copied to the device just prior to the crash and gave them a .REC file extension. I deleted them all and started the copy process from scratch.

The long and short of it is that after the more-or-less rock-solid performance I've had with the Cowon D2, the instability of the Archos 5 was a disappointment.

Conclusion

So, is the Archos 5 a good replacement for the Cowon D2? No.

Out of the box, the Archos 5 is slightly more capable than the D2 in terms of video (it can play my ripped DVDs just fine, with some caveats), but it is less capable in terms of audio. Only my MP3 and WAV audio test files played without trouble. This is a serious limitation for a device that bills itself as a media tablet. I also disliked having to purchase both an expensive plugin pack and the mini-dock just to get it up to the level of functionality it should have had right from the start.

That said, am I unhappy with my purchase? Not as much as one might think. True, it is not a replacement for my current favorite portable media player, but it is very useful in its own right, especially as a mobile node for my home media library (the parts that it can play). With any luck, many of the deficiencies will be erased as firmware updates are released.

In the meantime, my trusty D2 remains my hands-down favorite audio player.■

Daniel Bartholomew lives with his wife and children in North Carolina. His occasionally updated blog is at daniel-bartholomew.com, and he also can be found on Twitter as daniel_bart and on identi.ca (and Jaiku and Pownce) as bartholomew.

Resources

Archos: **www.archos.com**

GPL'd Source Code for Various Archos Products:
www.archos.com/support/support_tech/updates.html

Cowon D2: **www.cowonamerica.com/products/cowon/d2**

vTuner: **www.vtuner.com**

TECH TIP Slice and Dice PDF

Using poppler-tools and psutils, you can extract a range of pages from a larger PDF file. For example, if you want to extract pages 11–14 of the PDF file afile.pdf, you could use the following command:

```
$ pdftops afile.pdf - | psselect -p11-14 | ps2pdf - file-p11-14.pdf
```

The pdftops command converts the PDF file to PostScript; the psselect command selects the relevant pages from the PostScript, and the ps2pdf command converts the selected PostScript into a new PDF file.

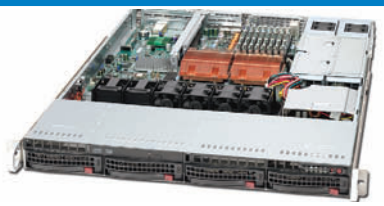
—JANOS GYERIK

The Straight Talk People℠
SINCE 1991

ABERDEEN

VMWARE CERTIFIED SOLUTIONS

ABERDEEN STIRLING 123



1U Dual Xeon VMware Certified Server
- Up to two Dual-Core or Quad-Core Intel® Xeon® processors
- Up to 32GB 667/533MHz Fully Buffered ECC DDR2 SDRAM
- Up to 4 x Hot-Swap SATA or SAS Hard Drives
- Universal I/O allows for 3 expansion cards in 1U
- Pre-installed VMware® ESXi on Disk-on-Module
- 650W High-efficiency Redundant Power Supply
- **5-Year Warranty**

Starting at **$1,999**

ABERDEEN STIRLING 229

2U Dual Xeon VMware Certified Server
- Up to two Dual-Core or Quad-Core Intel Xeon processors
- Up to 128GB 800/667/533MHz Fully Buffered ECC DDR2 SDRAM
- Up to 8 x 1TB (8.0TB) Hot-Swap SATA Hard Drives
- Up to 7 x Low-Profile Expansion Slots
- Pre-installed VMware ESXi on Disk-on-Module
- 700W High-efficiency Redundant Power Supply
- **5-Year Warranty**

Starting at **$2,425**

ABERDEEN STIRLING 428T

4U Dual Xeon VMware Certified Server
- Up to two Dual-Core or Quad-Core Intel Xeon processors
- Up to 64GB 667/533MHz Fully Buffered ECC DDR2 SDRAM
- Up to 8 x 1TB (8.0TB) Hot-Swap SATA Hard Drives
- Up to 6 x Full Height Expansion Slots
- Pre-installed VMware ESXi on Disk-on-Module
- 800W High-efficiency Redundant Power Supply
- **5-Year Warranty**

Starting at **$2,350**

ABERDEEN STIRLING 132T

1U Twin Node VMware Certified Server
- Up to two Dual-Core or Quad-Core Intel Xeon processors/node
- Twin Nodes allows for up to 4 processors & 16 cores in 1U
- Up to 64GB 800/667/533MHz Fully Buffered ECC DDR2/node
- Up to 2 x 1TB Hot-Swap SATA Hard Drives per node
- Pre-installed VMware ESXi on Disk-on-Module
- 980W High-efficiency Power Supply
- **5-Year Warranty**

Starting at **$2,880**

ABERDEEN STIRLING 244

2U Quad Xeon MP VMware Certified Server
- Up to four Quad-Core or Six-Core Intel Xeon MP processors
- Quad Six-Core allows for 24 processor cores in 2U
- Up to 192GB 667/533MHz Fully Buffered ECC DDR2 SDRAM
- Up to 6 x Hot-Swap SATA or SAS Hard Drives
- Pre-installed VMware ESXi on Disk-on-Module
- 1200W High-efficiency Redundant Power Supply
- **5-Year Warranty**

Starting at **$5,615**

ABERDEEN STIRLING 444

4U Quad Xeon MP VMware Certified Server
- Up to four Quad-Core or Six-Core Intel Xeon MP processors
- Quad Six-Core allows for 24 processor cores in 4U
- Up to 192GB 667/533MHz Fully Buffered ECC DDR2 SDRAM
- Up to 5 x Hot-Swap SATA or SAS Hard Drives
- Pre-installed VMware ESXi on Disk-on-Module
- 1200W High-efficiency Redundant Power Supply
- **5-Year Warranty**

Starting at **$6,625**

ABERDEEN ISCSI XDAS

DAS VMware Certified Expandable Storage
- IP SAN Solution
- Single or Redundant Controller
- Expandable up to 64TB in a single array
- 2U/12 Bay and 3U/16 Bay Models available
- SAS or SATA Hard Drive Support
- Fault-tolerant Modular Hardware Design
- **5-Year Warranty**

Starting at **$7,995**

ABERDEEN FIBRE XDAS

DAS VMware Certified Expandable Storage
- Hardware RAID5 and RAID6 engine by dedicated ASIC400
- Single or Redundant Controller
- Expandable up to 64TB in a single array
- 2U/12 Bay and 3U/16 Bay Models available
- SAS or SATA Hard Drive Support
- Fault-tolerant Modular Hardware Design
- **5-Year Warranty**

Starting at **$8,495**

888-297-7409
www.aberdeeninc.com/lj029

Dojo, Now with Drawing Tools!

Internalizing Dojo's "write once, deploy anywhere" philosophy, Dojo's gfx (pronounced "g-f-x" or sometimes "graphics") library packs a powerful 2-D drawing API that's capable of plugging in to an arbitrary renderer. Out of the box, it works with Canvas, Silverlight, SVG and VML, so regardless of which browser your application is ultimately viewed within, gfx has you covered.

MATTHEW RUSSELL

My article "Dojo: the JavaScript Toolkit with Industrial-Strength Mojo" in the July 2008 issue of *Linux Journal* illustrated how Dojo significantly lowers the amount of effort it takes to develop a cross-browser Web application by normalizing so many of the yucky aspects of Web programming, such as DOM manipulation, non-uniform aspects of the JavaScript across browsers, and repetitive tasks, such as styling nodes, performing AJAX requests and so forth. With that working knowledge, let's turn to Dojo's gfx library—a much more specialized aspect of the toolkit that's expressly designed to give you 2-D drawing tools that can be used to do anything from producing a cool-looking reflection of an image to creating an animated game to rendering a drag-and-drop graph.

So that you better understand exactly where gfx fits into the larger toolkit, recall that Dojo breaks down into roughly five components: Base, Core, Dijit, DojoX and Util. Base is the tiny dojo.js file that contains hard-live-without library code for common operations; Core includes most of the programmatic machinery for the toolkit; Dijit is an assortment of turnkey widgets; DojoX provides a collection of specialized subprojects; and Util provides a testing framework and scripts for tasks, such as minifying and consolidating JavaScript and CSS files. The gfx library is one of those many specialized subprojects that lives under the DojoX umbrella.

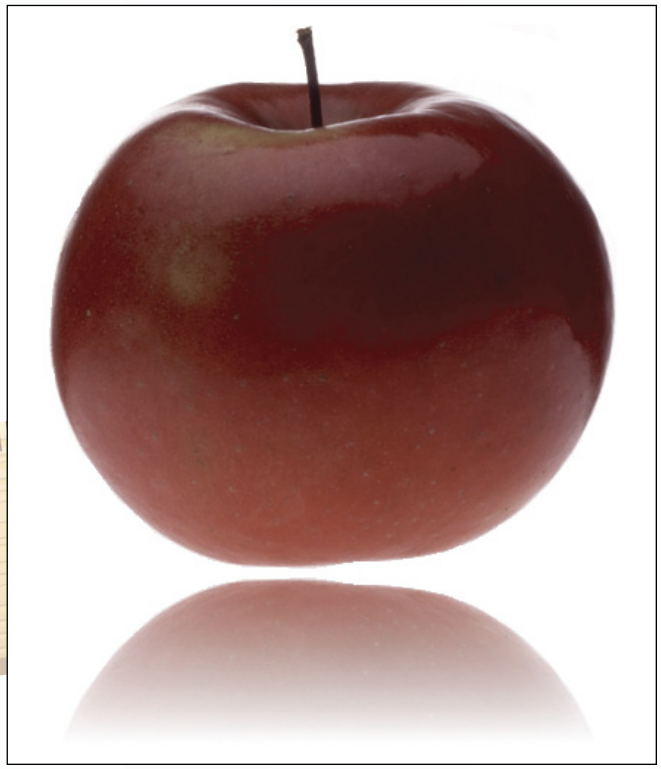


Figure 1. An example of a slick effect gfx can produce on an image.

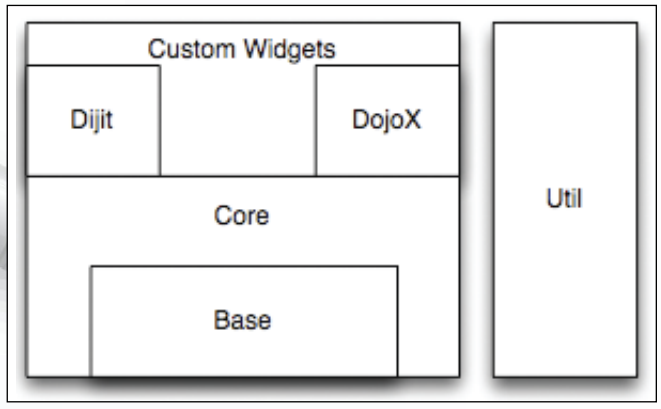


Figure 2. A Conceptual Portrayal of Dojo's Functional Architecture

NOTE:

A common misunderstanding is that everything within DojoX is experimental or necessarily unstable. Although there certainly are some alpha-quality subprojects within the DojoX namespace that you wouldn't want to rely on for long-term production scenarios, several DojoX subprojects (including gfx) are quite ready for mainstream use. In general, you should be able to check a project's README file to determine information about its status.

A Minimal Development Template

In order to demonstrate the various drawing concepts as clearly as possible, all of the examples you're about to see will plug right in to the following minimal HTML page. Although you're encouraged to download the entire toolkit eventually, so you have full access to the source code whenever you need it, let's take advantage of the version that's hosted on AOL's Content Delivery Network, as it's quicker to get up and running. The latest version of Dojo at the time of this writing is 1.2, so the minimal effort to put Dojo to work is the following page, which uses a script tag to cross-domain load the toolkit:

```
<html>
  <head>
       <title>Minimal Development Template</title>
       <script
            type="text/javascript"

src="http://o.aolcdn.com/dojo/1.2/dojo/dojo.xd.js">
       </script>
       <script type="text/javascript">
            dojo.addOnLoad(function() {
                 /*Add Dojo-dependent logic
                   here to avoid race conditions*/
            });
       </script>
  </head>
  <body>
  </body>
</html>
```

With the minimal template in place, it is trivial to load the gfx module and start drawing. The next section digs right in to various aspects of the API, but just so you can see where we're heading, consider the modification to the template that

Listing 1. A Minimal Drawing Example

```
<html>
  <head>
      <title>Square with a Diagonal Line</title>
      <script
          type="text/javascript"
          src="http://o.aolcdn.com/dojo/1.2/dojo/dojo.xd.js">
      </script>
      <script type="text/javascript">
          dojo.require("dojox.gfx");
          dojo.addOnLoad(function() {
              var node = dojo.byId("surface");
              var surface = dojox.gfx.createSurface(node, 600, 600);

              surface.createLine({
                  x1 : 0,
                  y1 : 0,
                  x2 : 600,
                  y2 : 600
              })
              .setStroke("black")
              ;
          });
      </script>
  </head>
  <body>
      <div style="width:600;height:600;border:solid 1px"
       id="surface"></div>
  </body>
</html>
```

instantiates a 600x600 pixel drawing surface and draws a line from the upper-left corner to the lower-right corner shown in Listing 1.

Although quite simple, the previous example taught us that the origin of the drawing surface is the upper-left corner with positive axes extending down and to the right, and that you can place a drawing surface into an arbitrary page element. Although not directly stated, the latter implies that you can have multiple drawing surfaces on a single page.

It's also worth noting that the style applied to the div element in no way applies to the gfx surface that is created. Internally, what happens is that the surface is created and placed inside of the div; thus, the containing div exhibits a 600x600 size with a visible border around it, and the surface that is placed into the div just so happened to be 600x600 pixels also. Without using Firebug to inspect the DOM, that may not have been obvious, so hopefully, mentioning it here avoids any confusion.

An additional aspect of this simple demonstration that's important to note is that the browser was detected and a default drawing renderer was assigned automatically without any special intervention. In the case of a Gecko- or KHTML-based browser, like Firefox or Konqueror, SVG is used as the default renderer; Internet Explorer defaults to VML.

Silverlight and Canvas can be configured to run on supported platforms via a gfxRenderer configuration switch supplied to djConfig via the script tag that loads Dojo into the page. For example, to instruct Firefox to use Canvas as the renderer you would provide the following script tag:

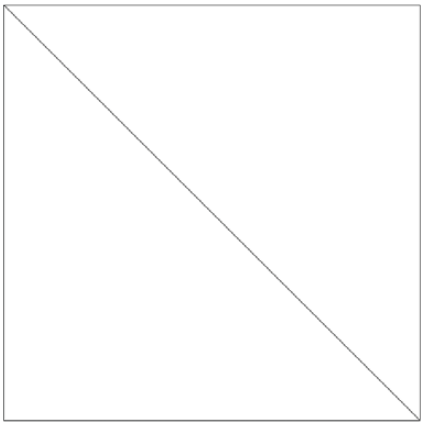


Figure 3. A 600x600 Drawing Surface with a Diagonal Line Drawn through It

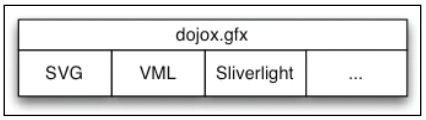


Figure 4. The gfx library's flexible design provides a uniform API that supplies a uniform abstraction on top of the most common drawing engines in the mainstream. Because it internally detects the drawing engine that's available, it works right out of the box.

```
<script
    type="text/javascript"
    djConfig="gfxRenderer:'canvas'"
    src="http://o.aolcdn.com/dojo/1.2/dojo/dojo.xd.js">
</script>
```

All Shapes and Sizes

The gfx API exposes a number of intuitive functions for common operations, such as creating rectangles, circles, lines, polylines and paths that are loosely based on the SVG standard as well as a set of custom attributes, such as stroke, fill color, rounded corners and more. Most of the

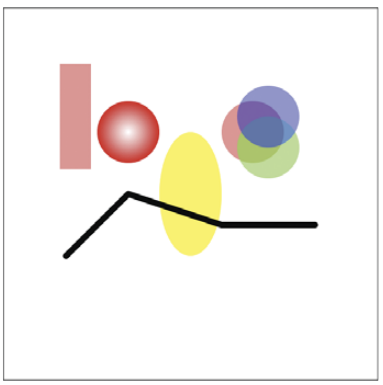


Figure 5. Dojo's fairly intuitive gfx API makes drawing a variety of customized elements easy and fun.

Mark your calendars. Plan your 2009 budgets.

7th Annual

2009 High Performance Linux on Wall Street Show and Conference

April 6, 2009 (Monday) **Roosevelt Hotel, NYC**
Madison Ave and 45th St, next to Grand Central Station

2009 High Performance Linux on Wall Street, April 6, with Low Latency Data Centers featured.

The success of High Performance on Wall Street this past September confirms the importance of a focused, convenient, no-hassle event where exhibitors and sponsors can meet their IT contacts under ideal circumstances at a one-day event.

Wall Street will continue to evolve with high speed, and low cost systems to support high volume data feeds and transaction activities worldwide.

These are two events in 2009 that will serve that need.

The 7th Annual Linux on Wall Street has been expanded to include **High Performance systems**, April 6, 2009, Roosevelt Hotel, NYC.

2008 Sponsors

IBM, Microsoft, Intel, Oracle, redhat, Enigmatec Corporation, Novell, Collaborative Software Initiative, QLogic

Conference Producer: A-TEAMGROUP

Show Management:
Flagg Management Inc
353 Lexington Ave, NY10016
(212) 286 0333
flaggmgmt@msn.com

Conference Management:
Pete Harris
(718) 237 2796
pete@a-teamgroup.com

Show Hours: Mon, Apr 6 8 - 4:00
Conference Hours: 8:30 - 4:50

Visit: www.linuxonwallstreet.com

6th Annual

High Performance Computing on Wall Street Show and Conference

September 14 (Monday) **Roosevelt Hotel, NYC**
Madison Ave and 45th St, next to Grand Central

The 2008 High Performance on Wall Street, Sept 22, was the most successful show and conference in our history.

Almost 1,000 IT managers from Wall Street and global financial firms attended the show and conference at the Roosevelt Hotel next to Grand Central Station. They came to hear Wall Street speakers and vendor experts talk about HPC, Virtualization, Cloud Computing, Low Latency, Grid, Blade, Cluster, HPC storage, HPC datacenters.

They also came to see HPC systems live and in operation on the show floor. They met vendor experts and received information to take to their offices for evaluation and purchase.

The 6th Annual High Performance Computing on Wall Street, Sept. 14, 2009 at the Roosevelt Hotel, will serve the HPC marketplace for Wall Street and the global financial markets.

Show Management:
Flagg Management Inc
353 Lexington Ave, NY10016
(212) 286 0333
flaggmgmt@msn.com

Conference Management:
Pete Harris
(718) 237 2796
pete@a-teamgroup.com

Conference Producer: A-TEAMGROUP

2008 Sponsors

Microsoft, Enigmatec Corporation, IBM, redhat, HP, Intel, Cisco, Sun microsystems, NetApp, Celoxica, Novell, Solace Systems, Supermicro, RapidMind, Blade Network Technologies, QLogic, Tervela, Data Synapse, APPRO, ClearSpeed, HPCwire, On-Demand Enterprise

Visit: www. highperformanceonwallstreet.com

Listing 2. An Assortment of Shapes

```
dojo.addOnLoad(function() {
    var node = dojo.byId("surface");
    var surface = dojox.gfx.createSurface(node, 600, 600);

    surface.createEllipse({
        cx : 300,
        cy : 300,
        rx  : 50,
        ry  : 100
    })
    .setFill("yellow")
    ;

    surface.createRect({
        x : 90,
        y : 90,
        width : 50,
        height : 170
    })
    .setFill([255,0,0,0.5])
    ;

    surface.createCircle({
        cx : 400,
        cy : 200,
        r  : 50
    })
    .setFill([255,0,0,0.5]);

    surface.createCircle({
        cx : 425,
        cy : 225,
        r  : 50
    })
    .setFill([0,255,0,0.5])
    ;

    surface.createCircle({
        cx : 425,
        cy : 175,
        r  : 50
    })
    .setFill([0,0,255,0.5])
    ;

    surface.createPolyline([
            100,400,
            200,300,
            350,350,
            500,350
    ])
    .setStroke({
        width : 10,
        join : "round",
        cap : "round"
    })
    ;

    surface.createCircle({
        r : 50,
        cx : 200,
        cy: 200
    })
    .setFill({
        type: "radial",
        cx : 200,
        cy: 200,
        r:50,
        colors: [
            {color:"white",offset:0},
            {color:"red",offset:1}]
    })
    ;
});
```

methods support "chaining syntax", which allows you to operate on the results of the previous operation repeatedly, leading to crisp code and clean syntax, so long as you do not abuse the device (Listing 2).

Hopefully, the code mostly speaks for itself. The various types of objects that you can create are usually framed in the same way that they are presented in grade school. For example, a circle has a center point and a radius defined by cx, cy and r. Given a circle, you could set a fill color in a number of different ways: a string value, an rgb(a) tuple or even something more complex like a radial gradient with custom parameters of its own.

3x3 Matrix Transforms

Using a well-designed API with nice mnemonic devices is useful for much of the routine drawing you'll be doing, but what about when you need to do something a lot more in depth? Although this is where a lot of JavaScript graphics libraries fall short, gfx absolutely shines here by equipping you with the ability to perform arbitrary 3x3 matrix transformations.

Just in case you don't have a background with graphics, it may not be immediately apparent how 3x3 matrices and "all of that math" is useful. Basically, 3x3 matrices provide a compact way to express the three common operations that you do with objects all at the same time:

- Translation: adjusting the position of an object in the x and y directions.
- Rotation: adjusting the position of an object in the clockwise or counterclockwise directions usually (but not necessarily) around its center point.
- Scaling: adjusting the size of an object by a scalar multiplier.

Don't freak out quite yet if you're not a math buff and don't want to sink time into re-learning linear algebra just to get started with that great idea you had for a game or drawing application. Many of the common operations for manipulating shapes come with intuitive wrappers. To illustrate

a trivial example, let's assume that you want to draw a square but then rotate it around its center point so that it looks like a diamond:

```
dojo.addOnLoad(function() {
    var node = dojo.byId("surface");
    var surface = dojox.gfx.createSurface(node, 600, 600);

    rect1 = surface.createRect({
        x: 200,
        y: 200,
        width : 200,
        height:200
    })
    .setFill("red")
    .setTransform([dojox.gfx.matrix.rotategAt(45,300,300)])
    ;
});
```

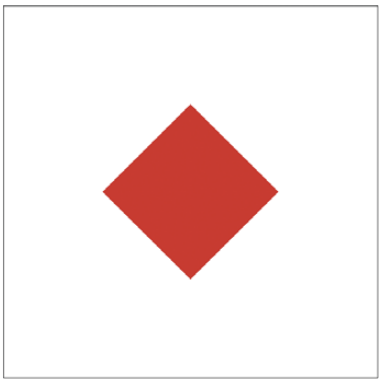


Figure 6. The effect of drawing a square and rotating 45 degrees around its center point.

With an upper-left corner at point (200,200) and a width and height of 200 pixels, the square originally was centered on the surface. Then, applying a 45-degree rotation around the square's center point of (300,300) rotated it in place.

To illustrate the effect of successively applying transformation matrices, let's draw the very same diamond but rely on explicit translation to position it in the center of the surface before rotating it versus positioning it via the createRect function:

```
dojo.addOnLoad(function() {
    var node = dojo.byId("surface");
    var surface = dojox.gfx.createSurface(node, 600, 600);

    rect1 = surface.createRect({
        /* x and y default to (0,0) */
        width : 200,
        height:200
    })
    .setFill("red")
    .setTransform([
        dojox.gfx.matrix.translate(200,200),
        dojox.gfx.matrix.rotategAt(45,100,100)
    ])
    ;
});
```

In general, it is immensely more convenient to draw most shapes initially in a coordinate system with perpendicular x and y axes and then apply final positioning via translation and rotation. An important technicality to be aware of with successive transformations, however, is that the order in which the transforms are applied does matter, and the original position of the object is normally the point of reference. For instance, in the previous example, the shape explicitly was translated as 200 pixels in the x and y directions, but its original center point from before the translation is applied serves as the basis of rotation.

If you're unconvinced that a shape as simple as a diamond would benefit much from the convenience of matrix transforms, just consider the extra work involved in calculating the exact coordinates for its corners, and you'll quickly see that it's easier to reason about "rotated squares" than it is about "native diamonds".

Manipulating Groups

It won't be long before you'll find that it's far more convenient to transform entire groups of objects instead of applying individual transforms to each object in the group. Let's consider the task of drawing a simple arrow that is nothing more than a line with a triangle on the end of it. Although you could use a path to construct the entire arrow, take a look at how groups can be useful by combining the results from the createLine function and the createPath function (Listing 3).

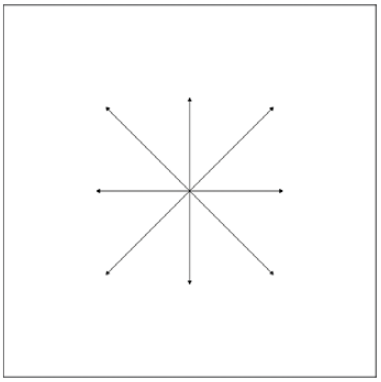


Figure 7. It's generally a bit easier to think in terms of objects that have been rotated than trying to determine exact coordinates for shapes that don't fit nicely into a simple perpendicular frame of reference.

Attempting to calculate the three points for each of the arrows without the benefit of rotation quickly demonstrates just how laborious high-school geometry really can be; perhaps putting it to work with gfx makes it at least a little more interesting.

Drag-and-Droppable Dominoes

Because it's so common to want to interact with graphics, Dojo's gfx library has gone a long way to do most of the legwork for you in this use case as well. To wrap up some aspects of drawing, let's put together a little demonstration that draws a domino on the screen and then add drag-and-drop capabilities to it. As you're about to see, the laborious part of the effort is actually drawing something interesting enough that you'd

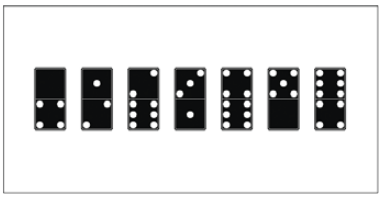


Figure 8. With the logic to draw the drag-and-droppable dominoes in place, now all that's left is to write some game logic. (An exercise for the most interested of readers.)

Listing 3. Arrows in All Four Quadrants

```
dojo.addOnLoad(function() {
    var node = dojo.byId("surface");
    var surface = dojox.gfx.createSurface(node, 600, 600)

    function drawArrow(p) {
        ////////////////////////////////////////////////////
        //Create a group that can be manipulated as a whole
        ////////////////////////////////////////////////////
        var group = surface.createGroup();

        var x1 = p.start.x,
            y1=p.start.y,
            x2 = p.end.x,
            y2=p.end.y;

        var len = Math.sqrt(Math.pow(x2-x1,2) + Math.pow(y2-y1,2));

        var _defaultStroke = {
            color : "black",
            style : "solid",
            width : 1
        };

        ////////////////////////////
        //Add a line to the group
        ////////////////////////////
        group.createLine({
            x1 : 0,
            y1 : 0,
            x2 : 0+len,
            y2 : 0
        })
        .setStroke(p.stroke || _defaultStroke)
        ;

        var _arrowHeight = p.arrowHeight || 5;
        var _arrowWidth = p.arrowWidth || 3;

        ////////////////////////////////////////////////////
        //Add a custom path that is a triangle to the group
        ////////////////////////////////////////////////////
        group.createPath()
        .moveTo(len-_arrowHeight,0)
        .lineTo(len-_arrowHeight,-_arrowWidth)
        .lineTo(len,0)
        .lineTo(len-_arrowHeight,_arrowWidth)
        .lineTo(len-_arrowHeight,0)
        .setStroke(p.stroke || _defaultStroke)
        .setFill(p.stroke ? p.stroke.color : "black" )
        ;

        var _rot = Math.asin((y2-y1)/len)*180/Math.PI;
        if (x2 <= x1) {_rot = 180-_rot;}

        ////////////////////////////////////////////////////////////
        //Translate and rotate the entire group as a whole
        ////////////////////////////////////////////////////////////
        group.setTransform([
            dojox.gfx.matrix.translate(x1,y1),
            dojox.gfx.matrix.rotategAt(_rot,0,0)
        ]);
    }

    //diagonals
    drawArrow({start: {x:300,y:300}, end: {x : 435, y : 435}});
    drawArrow({start: {x:300,y:300}, end: {x : 165, y : 165}});
    drawArrow({start: {x:300,y:300}, end: {x : 435, y : 165}});
    drawArrow({start: {x:300,y:300}, end: {x : 165, y : 435}});

    //up, down, left, right
    drawArrow({start: {x:300,y:300}, end: {x : 300, y : 450}});
    drawArrow({start: {x:300,y:300}, end: {x : 300, y : 150}});
    drawArrow({start: {x:300,y:300}, end: {x : 150, y : 300}});
    drawArrow({start: {x:300,y:300}, end: {x : 450, y : 300}});
});
```

actually want to drag and drop it. The actual mechanics of making it drag-and-droppable amounts to *one whole line of code* (Listing 4—note that the full code for Listing 4 is available on *LJ*'s FTP site; see Resources).

Charting: gfx on Steroids

Perhaps the ultimate test of an API is a few good examples of what you can build with it. One of the ultimate demonstrations of gfx's flexibility and power is Dojo's charting library, another DojoX subproject. A comprehensive introduction of the charting library would entail an article of its own, so until that time comes, you can find some great documentation on Dojo charting from the Dojo Key Links page. And, of course, you always can read over the source, which is located in the dojox.charting module of the toolkit's source code, if you want to get an idea of how much work goes into aligning labels, drawing tick marks and so on.

In addition to equipping you with many of the basic charts you'd want to use in a Web application, charting recently got

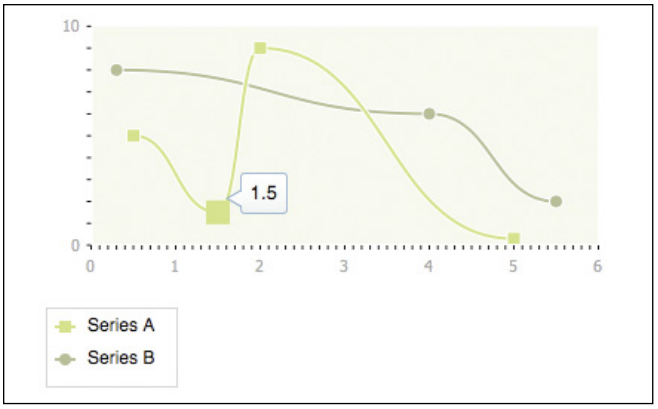


Figure 9. An example of the charts you can draw with Dojo—no Flash required!

a boost with a number of cool new features, including event support so that custom tooltips and animations can occur

Listing 4. Drag-and-Drappable Dominoes

```
<html>
  <head>
    <title>Dominoes!</title>
    <script type="text/javascript"
        src="http://o.aolcdn.com/dojo/1.2/dojo/dojo.xd.js">
    </script>
    <script type="text/javascript">
        dojo.require("dojox.gfx");
        dojo.require("dojox.gfx.move");
        dojo.addOnLoad(function() {
            var node = dojo.byId("surface");
            var surface = dojox.gfx.createSurface(node, 600, 300);

            /* Using sane ratios for layout, construct a domino */
            function drawDomino(surface,x,y,num1,num2,_width) {
                var surface = surface.createGroup();
                var _width = _width || 200;
                var _height = 2*_width, _r = _width/20;

                //draw an empty domino...
                var rect1 = surface.createRect({
                    x : x,
                    y : y,
                    width : _width,
                    height : _height,
                    r : _r
                })
                .setStroke("black")
                .setFill("black")
                ;
                var rect2 = surface.createRect({
                    x : x+ _r/2,
                    y : y+ _r/2,
                    width : _width -_r,
                    height :_height -_r,
                    r : _r
                })
                .setStroke({width: _r/4, color: "white"})
                ;
                var line = surface.createLine({
                    x1 : x + _r/2,
                    y1 : y+_height/2,
                    x2 : x+ _width - _r/2,
                    y2 : y+_height/2,
                })
                .setStroke({width: _r/4, color: "white"})
                ;
                //numbers 1,3,5 have dots in the center
                if (num1 == 1 || num1 == 3 || num1 == 5) {
                    surface.createCircle({
                        cx : x+_width/2,
                        cy : y+_height/4,
                        r : _width/10
                    })
                    .setStroke("white")
                    .setFill("white")
                    ;
                }
                if (num2 == 1 || num2 == 3 || num2 == 5) {
                    surface.createCircle({
                        cx : x+_width/2,
                        cy : y+_height/4*3,
                        r : _width/10
                    })
                    .setStroke("white")
                    .setFill("white")
                    ;
                }
                //numbers >= 2 have two of the corners filled in
                if (num1 >= 2) {
                    surface.createCircle({
                        cx : x+_width/6*5,
                        cy : y+_height/12,
                        r : _width/10
                    })
                    .setStroke("white")
                    .setFill("white")
                    ;
                    surface.createCircle({
                        cx : x+_width/6,
                        cy : y+_height/12*5,
                        r : _width/10
                    })
                    .setStroke("white")
                    .setFill("white")
                    ;
                }

                /*** SNIP - Go to LJ FTP site to download
                     the rest of this code (see Resources) ***/

                return surface;
            }
            var width=50,
                padding=50;

            for (var i=0; i <= 6; i++) {
                var d = drawDomino(
                    surface,
                    i*75+padding,
                    2*padding,
                    i,
                    Math.floor(Math.random()*7),
                    width
                );
                //Make the group drag-and-droppable
                new dojox.gfx.Moveable(d);
            }
            //Adjust the z-index so last domino is on top
            dojo.subscribe("/gfx/move/start", function(m) {
                m.shape.moveToFront();
            });
        })
    </script>
  </head>
  <body>
    <div id="surface"
     style="position:absolute;width:600;height:300;border:solid1px;">
    </div>
  </body>
</html>
```

Listing 5. Preview of Dojo's Charting API

```
////////////////////////////////////////////////////
// This function demonstrates the general form of
// putting Dojo's charting API built on top of
// gfx to use. Pass in a node, customize the chart,
// and let Dojo take care of the rest
////////////////////////////////////////////////////
new dojox.charting.Chart2D(node))
    .setTheme(dojox.charting.themes.PlotKit.blue)
    .addPlot("default", {
        type: "Default",
        lines: true,
        markers: true,
        tension: 2
    })
    .addAxis("x", {
        min: 0,
        max: 6,
        majorTick: { stroke: "black", length: 3 },
        minorTick: { stroke: "gray", length: 3 }
    })
    .addAxis("y", {
        vertical: true,
        min: 0,
        max: 10,
        majorTick: { stroke: "black", length: 3 },
        minorTick: { stroke: "gray", length: 3 }
    })
    .addSeries("Series A", [
        { x: 0.5, y: 5 },
        { x: 1.5, y: 1.5 },
        { x: 2, y: 9 },
        { x: 5, y: 0.3 }
    ])
    .addSeries("Series B", [
        { x: 0.3, y: 8 },
        { x: 4, y: 6 },
        { x: 5.5, y: 2 }
    ])
    .render()
    ;
```

within charts—that kind of visual flair makes all the difference. To give you an idea of just how easy the charting API is to get up and running, consider the code blurb in Listing 5 that shows how to create a chart.

Although only a teaser, it's worthwhile to note the charting API focus on charting—not on raw drawing operations—so you can focus on the semantics of the task at hand instead of the implementation details associated with lower-level operations. In general, you simply provide some data that says what kind of chart you'd like, how to customize the axes and pass in the series data. Setting up event handlers, legends and other related things all work much the same way.

There's Plenty More Where That Came From

2-D drawing is an enormous topic in and of itself, and no single article could cover all the nooks and crannies adequately. This article is designed to give you an idea of just how easy Dojo makes 2-D for the Web, which hopefully motivates you to start experimenting with the examples and check out the API docs.■

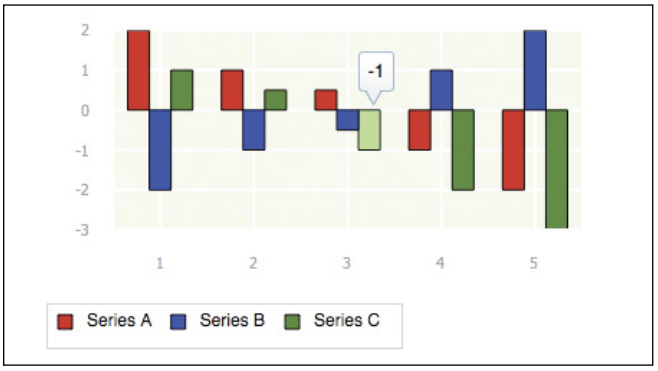


Figure 10. Another Example Chart Drawn with Dojo

Matthew Russell is an open Web technology consultant and the author of *Dojo: The Definitive Guide* (O'Reilly, June 2008).

Resources

Full Drag-and-Droppable Dominoes Code (Listing 4): **ftp.linuxjournal.com/pub/lj/issue178/10308.tgz**

"Dojo: the JavaScript Toolkit with Industrial-Strength Mojo" by Matthew Russell, *LJ*, July 2008: **www.linuxjournal.com/article/9900**

The Dojo Toolkit: **dojotoolkit.org**

The gfx API Documentation: **docs.google.com/View?docid=d764479_1hnb2tn**

Reflection with gfx: **dojotdg.com/2008/09/a-simple-degradable-reflection-widget**

Overview of Transformation Matrices: **en.wikipedia.org/wiki/Transformation_matrix**

Custom Fonts with gfx: **www.sitepen.com/blog/2008/09/08/custom-fonts-with-dojoxgfx**

Dojo: The Definitive Guide (O'Reilly, June 2008): **oreilly.com/catalog/9780596516482**

Dojo: The Definitive Guide (Official Online Compendium): **dojotdg.com**

Dojo Campus: **dojocampus.org**

Dojo Key Links: **dojotoolkit.org/key-links**

Dojo Foundation: **dojofoundation.org**

SXSW '09 interactive

march 13–17 • austin, tx

SXSW INTERACTIVE FESTIVAL: CONNECT, DISCOVER, INSPIRE

Attracting digital creatives and new media entrepreneurs, the 15th annual South by Southwest (SXSW) Interactive Festival gives you both practical how-to information as well as unparalleled career inspiration. Attend this legendary gathering of the tribes to renew your link to the cutting edge.

Tony Hsieh to Deliver Opening Remarks on Saturday, March 14

At Zappos.com, Tony Hsieh has fostered a culture where extraordinary customer service is the norm. On Saturday, March 14, hear him talk about how good deeds can help you leverage the power of your audience to massively extend your brand.

Scheduled 2009 panels include:

Appfrica: How Web Applications Are Helping Emerging Markets Grow • Being a UX Team of One • CSS3: What's Now, What's New and What's Not? • Gestural UI: iPhone Taught Us Flick and Pinch. What's Next? • Get Me Rewrite! Developing APIs and the Changing Face of News • How To Roll Your Own API • The Invisible Web and Ubiquitous Computing • Make it So (Sexy): Lustful Design in Mainstream Science Fiction • Making Web Widgets Accessible: Tools and Techniques • More Secrets of JavaScript Libraries • OpenID, OAuth, Data Portability and the Enterprise • Post Standards: Creating Open Source Specs • Version Control: No More Save As... Many, many, more to be announced!

ScreenBurn Panels & Evening Events
March 13-17, 2009

NEW!

ScreenBurn Arcade
Friday, March 13 • 2-6pm
Saturday, March 14 • 12-6pm
ScreenBurn Arcade EXTRA DAY!
Sunday, March 15 • 12-6pm

REGISTER TO ATTEND SXSW INTERACTIVE 2009

Register before **January 16** to receive the next discounted rate and get the best choice of available hotels: **sxsw.com/attend**

Attend SXSW Film and SXSW Interactive at a bargain rate by purchasing a **Gold Badge**

SOUTH BY SOUTHWEST INTERACTIVE FESTIVAL
March 13-17, 2009 | Austin, Texas | **sxsw.com**

Miller Lite · FUZE · Microsoft Silverlight · The Austin Chronicle austinchronicle.com

Web 2.0 Development with the Google Web Toolkit

The Google Web Toolkit allows for modern Web development using Java, without ever needing to write a single line of HTML or JavaScript.

FEDERICO KEREKI

There's much hype related to Web 2.0, and most people agree that software like Google Maps, Gmail and Flickr fall into that category. Wouldn't you like to develop similar programs allowing users to drag around maps or refresh their e-mail inboxes, all without ever needing to reload the screen?

Until recently, creating such highly interactive programs was, to say the least, difficult. Few development tools, little debugging help and browser incompatibilities all added up to a complex mix. Now, however, if you want to produce such cutting-edge applications, you can use modern software methodologies and tools, work with the high-level Java language, and forget about HTML, JavaScript and whether Firefox and Internet Explorer behave the same way. The Google Web Toolkit (GWT) makes it easy to do a better job and produce more modern Web 2.0 programs for your users.

What Is Web 2.0?

This question has several answers, including Sir Tim Berners-Lee's (the creator of the World Wide Web) view that it's just a reuse of components that were there already. It originally was coined by Tim O'Reilly, promoting "the Web as a platform", with data as a driving force and technologies fostering innovation by assembling systems and sites that get information and features from distributed, different, independent developers and services.

This notion goes along with the idea of letting users run applications entirely through a browser, without installing anything on their machines. These new programs usually feature rich, user-friendly interfaces, akin to the ones you would get from an installed program, and they generally are achieved with AJAX (see the What Is AJAX? sidebar) to reduce download times and speed up display time.

Web 2.0 applications use the same infrastructure that developers are largely already familiar with: dynamic HTML, CSS and JavaScript. In addition, they often use XML or JSON for representing and communicating data between the server and browser. This data communication is often done using Web service requests via the DOM API XMLHttpRequest.

What Is the Google Web Toolkit?

The Google Web Toolkit (GWT—rhymes with "nitwit") is a tool for Web programmers. Its first public appearance was in May 2006 at the JavaOne conference. Currently (at the time of this writing), version 1.5.3 has just been released. It is licensed mainly under the Apache 2.0 Open Source License, but some of its components are under different licenses. Don't confuse JavaScript with Java; despite the name, the languages are unrelated, and the similarities come from some common roots.

In short, GWT makes it easier to write high-performing, interactive, AJAX applications. Instead of using the JavaScript language (which is powerful, but lacking in areas like modularity and testing features, making the development of large-scale systems more difficult), you code using the Java language, which GWT compiles into optimized, tight JavaScript code. Moreover, plenty of software tools exist to help you write Java code, which you now will be able to use for testing, refactoring, documenting and reusing—all these things have become a reality for Web applications.

You also can forget about HTML and DHTML (Dynamic HTML, which implies changing the actual source code of the page you are seeing on the fly) and some additional subtle compatibility issues therein. You code using Java widgets (such as text fields, check boxes and more), and GWT takes care of converting them into basic HTML fields and controls. Don't worry about localization matters either; with GWT, it's easy to produce locale-specific versions of code.

There's another welcome bonus too. GWT takes care of the differences between browsers, so you don't have to spend time writing the same code in different ways to please the particular quirks of each browser. Typically, if you just code away and don't pay attention to those small details, your site will end up looking fine in, say, Mozilla Firefox, but won't work at all in Internet Explorer or Safari. This is a well-known classic Web development problem, and it's wise to plan for compatibility tests before releasing any site. GWT lets you forget about those problems and focus on the task instead.

What Is AJAX?

The standard model for Web applications is something like this: you get a screenful of text and fields from a server, you fill in some fields, and when you click a button, the browser sends the data you typed to a server (wait), which processes it (wait), and sends back an answer (wait), which your browser displays, and then the cycle restarts. This is by far the most common way Web applications operate, and you must get used to the delays. Nothing happens immediately, because every answer that needs data from a server requires a round trip.

AJAX (Asynchronous JavaScript And XML) is a technique that lets a Web application communicate in the background (asynchronously) with a Web server to exchange (send or receive) data with it. This does away with the requirement to reload the whole page after every action or user click. Thus, using AJAX increases the level of interaction, does away with waiting for pages to reload and allows for enhanced functionality. A well-programmed application will send requests in the background, as you are doing other things, so you won't have to stare at a blank screen or a turning-hourglass cursor. This is the Asynchronous part of AJAX acronym.

The next part of the AJAX acronym is JavaScript. JavaScript allows a Web page to contain a program, and this program is what allows the Web page to connect to a server as previously described. However, it's not just a question of having JavaScript, but also of how it is implemented in the browser. Both Firefox and Internet Explorer both provide AJAX access, but with some differences, so programmers must take those differences into account when doing the connection. Data is usually retrieved using XMLHttpRequest, but other techniques are possible, such as using iframes.

Finally, the last part of the AJAX acronym is XML. XML is a standard markup language, used for sharing and passing information. As we've seen, the name of the DOM API for making Web service requests is named XMLHttpRequest, and most likely, the original intent was that XML be used as the protocol for exchanging data between browser and server. However, neither the X in AJAX nor the XML in XMLHttpRequest means that you have to use XML; any data protocol at all, including no protocol, can be used.

JSON (JavaScript Object Notation) often is used; it's more lightweight than XML, and as you might guess by its name, is often a better fit for JavaScript. See Figure 3 for some actual JSON code; remember, it's not meant to be clear to humans, but compact and easy to understand for machines.

AJAX comprises basic technologies that have been around for a while now, and the AJAX term itself was created in 2005 by Jesse Garrett. GWT uses AJAX to allow the client program to communicate with the server or execute procedures on it in a fully transparent way. Of course, you also can use AJAX explicitly for any special purposes you might have.

According to its developers, GWT produces high-quality code that matches (and probably surpasses) the quality (size and speed) of handwritten JavaScript. The GWT Web page contains the motto "Faster AJAX than you can write by hand!"

GWT also endeavors to minimize the resulting code size to speed up transfers and shorten waiting time. By default, the end code is mostly unreadable (being geared toward the browser, not a snooping user), but if you have any problems, you can ask for more legible code so you can understand the relationship between your Java code and the produced JavaScript.

Getting Started with GWT

Before installing GWT, you should have a few things already installed on your machine:

- Java Development Kit (JDK), so you can compile and test Java applications; several more tools also are included.
- Java Runtime Environment (JRE), including the Java Virtual Machine (JVM) and all the class libraries required for production and development environments.
- A development environment—Google's own developers use Eclipse, so you might want to follow suit. Or, you can install GWT4NB and do some tweaking and fudging and work with NetBeans, another popular development environment.

GWT itself weighs in at about 27MB; after downloading it, extract it anywhere you like with `tar jxf ../gwt-linux-1.5.3.tar.bz2`. No further installation steps are required. You can use GWT from any directory.

For this article, I used Eclipse. For more serious work, you probably also will require some other additions, such as the Data Tools Platform (DTP), Eclipse Java Development Tools (JDT), Eclipse Modeling Framework (EMF) and Graphical Editing Framework (GEF), but you easily can add those (and more) with Eclipse's own software update tool (you can find it on Eclipse's main menu, under Help—and no, I don't know why it is located there).

Before starting a project, you should understand the four components of GWT:

- When you are developing an application, GWT runs in hosted mode and provides a Web browser (and an embedded Tomcat Web server), which allows you to test your Java application the same way your end users would see it. Note that you will be able to use the interactive debugging facilities of your development suite, so you can forget about placing alert() commands in JavaScript code.
- To help you build an interface, there is a Web interface library, which lets you create and use Web browser widgets, such as labels, text boxes, radio buttons and so on. You will do your Java programming using those widgets, and the compilation process will transform them into HTML-equivalent ones.
- Because what runs in the client's browser is JavaScript, there needs to be a Java emulation library, which provides JavaScript-equivalent implementations of the most common Java standard classes. Note that not all of Java is available, and there are restrictions as to which classes you can use. It's possible that you will have to roll your own code if you want to use an unavailable class. As of version 1.5, GWT covers much of the JRE. In addition, as of version 1.5, GWT supports using Java 5.
- Finally, in order to deploy your application, there is a Java-to-JavaScript compiler (translator), which you will use to produce the final Web code. You will need to place the resulting code, the JavaScript, HTML and CSS on your Web server later, of course.

If you are like most programmers, you probably will be wondering about your converted application's performance. However, GWT generates ultra-compact code that can be compressed and cached further, so end users will

Packages

Using GWT requires learning about several packages. The most important ones are:

- com.google.gwt.http.client: provides the client-side classes for making HTTP requests and processing the received responses. You will use it if you need to do some AJAX on your own, beyond the calls done by GWT itself.
- com.google.gwt.i18n.client: provides internationalization support. You will need it if you are developing a system that will be available in several languages.
- com.google.gwt.json.client and com.google.gwt.xml.client: used for parsing and reading XML and JSON data.
- com.google.gwt.junit.client: used for building automated JUnit tests.
- com.google.gwt.user.client.ui: provides panels, buttons, text boxes and all the other user-interface elements and classes. You certainly will use these.
- com.google.gwt.user.client.rpc and com.google.gwt.user.server.rpc: these have to do with remote procedure calls (RPCs). GWT allows you to call server code transparently, as if the client were residing in the same machine as the server.

You can find information on these and other packages on-line, at **google-web-toolkit.googlecode.com/svn/javadoc/1.5/index.html**.

download a few dozen kilobytes of end code, only once. Furthermore, with version 1.5, the quality of the generated code is approaching (and even surpassing) the quality of handwritten JavaScript, especially for larger projects. Finally, because you won't need to waste time doing debugging for every existing Web browser, you will have more time for application development itself, which lets you produce more features and better applications.

A GWT Example

Now, let's turn to a practical example. Creating a new project is done with the command line rather than from inside Eclipse. Create a directory for your project, and `cd` to it. Then create a project in it, with:

```
/path/to/GWT/projectCreator -eclipse ProjectName
```

Next, create a basic empty application, with:

```
/path/to/GWT/applicationCreator -eclipse ProjectName \
        com.CompanyName.client.ApplicationName
```

Then, open Eclipse, go to File→Import→General, choose Existing Projects into Workspace, and select the directory in which you created your project. Do not check the Copy Projects into Workspace box so that the project will be left at the directory you created.

After doing this, you will be able to edit both the HTML and Java code, add new classes and test your program in hosted mode, as described earlier. When you are satisfied with the final product, you can compile it (an appropriate script was generated when you created the original project) and deploy it to your Web server.

Let's do an example mashup. We're going to have a text field, the user will type something there, and we will query a server (okay, with only one server, it's not much of a mashup, but the concept can be extended easily) and show the returned data. Of course, for a real-world application, we wouldn't display the raw data, but rather do further processing on it. The example project itself will be called exampleproject, and its entry point will be *example*, see Listing 1 and Figure 1.

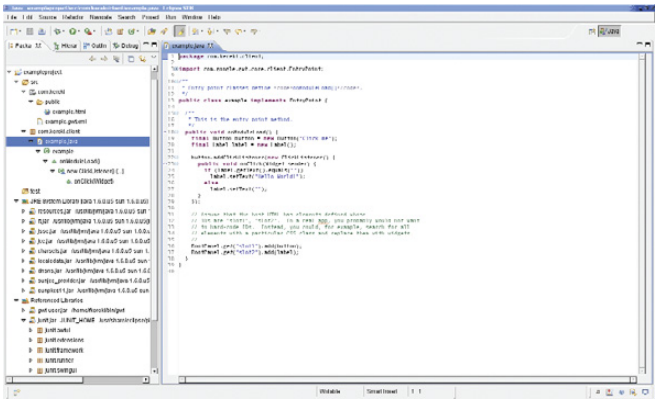


Figure 1. The recently imported project—the code just shows a welcome message.

Listing 1. Projects must be created by hand, outside Eclipse, and imported into it later.

```
# cd
# md examplefiles
# cd examplefiles
# ~/bin/gwt/projectCreator -eclipse exampleproject
Created directory ~/examplefiles/src
Created directory ~/examplefiles/test
Created file ~/examplefiles/.project
Created file ~/examplefiles/.classpath

# ~/bin/gwt/applicationCreator -eclipse exampleproject \
       com.kereki.client.example
Created directory ~/examplefiles/src/com/kereki
Created directory ~/examplefiles/src/com/kereki/client
Created directory ~/examplefiles/src/com/kereki/public
Created file ~/examplefiles/src/com/kereki/example.gwt.xml
Created file ~/examplefiles/src/com/kereki/public/example.html
Created file ~/examplefiles/src/com/kereki/client/example.java
Created file ~/examplefiles/example.launch
Created file ~/examplefiles/example-shell
Created file ~/examplefiles/example-compile
```

According to the Getting Started instructions on the Google Web Toolkit site, you should click the Run button to start running your project in hosted mode, but I find it more practical to run it in debugging mode. Go to Run→Debug, and launch your application. Two windows will appear: the development shell and the wrapper HTML window, a special version of the Mozilla browser. If you do any code changes, you won't have to close them and relaunch the application. Simply click Refresh, and you will be running the newer version of your code.

Now, let's get to our changes. Because we're using JSON and HTTP, we need to add a pair of lines:

```
<inherits name='com.google.gwt.json.JSON'/>
```

and:

```
<inherits name='com.google.gwt.http.HTTP'/>
```

to the example.gwt.xml file. We'll rewrite the main code and add a couple packages to do calls to servers that provide JSON output (see The Same Origin Policy sidebar). For this, add two classes to the client: JSONRequest and JSONRequestHandler; their code is shown in Listings 2 and 3.

Let's opt to create the screen completely with GWT code. The button will send a request to a server (in this case, Yahoo! News) that provides an API with JSON results. When the answer comes in, we will display the received code in a text area. The complete code is shown in Listing 4, and Figure 3 shows the running program.

After testing the application, it's time to distribute it. Go to the directory where you created the project, run the compile script (in this case, example_script.sh), and copy the resulting files to your server's Web pages directory.

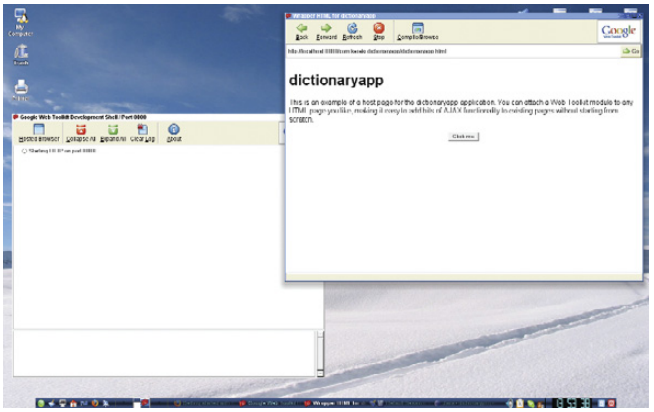


Figure 2. Running the Created Application the First Time, in Hosted Mode

The Same Origin Policy

The Same Origin Policy (SOP) is a security restriction, which basically prevents a page loaded from a certain origin to access a page from a different origin. By origin, we mean the trio: protocol + host + port. In http://www.mysite.com:80/some/path/to/a/page, the protocol is http, the host is www.myhost.com, and the port is 80. The SOP would allow access to any document coming from http://www.mysite.com:80, but disallow going to https://www.mysite.com:80/something (different protocol), http://dev.mysite.com:80/something (different host) or http://www.mysite.com:81/something (different port).

Why is this a good idea? Without it, it would be possible for JavaScript from a certain origin to access data from another origin and manipulate it secretly. This would be the ultimate phishing. You could be looking at a legitimate, valid, true page, but it might be monitored by a third party. With SOP in place, you know for certain that whatever you are viewing was sent by the true origin. There can't be any code from other origins.

Of course, for GWT, this is a bit of a bother, because it means that a client application cannot simply connect to any other server or Web service to get data from it. There are (at least) two ways around this: a special, simpler way that allows getting JSON data only or a more complex solution that implies coding a server-side proxy. Your client calls the proxy, and the proxy calls the service. Both solutions are explained in the *Google Web Toolkit Applications* book (see Resources). In this article, we use the JSON method, and you can find the source code at **www.gwtsite.com/code/webservices**.

The simple JSON method requires a special callback routine, and this could be a showstopper. However, many sites implement this, including Amazon, Digg, Flickr, GeoNames, Google, Yahoo! and YouTube, and the method is catching on, so it's quite likely you will be able to find an appropriate service.

Listing 2. Source Code for the JSONRequest Class

```
package com.kereki.client;
public class JSONRequest {
    public static void get(String url,
                           JSONRequestHandler handler) {
        String callbackName = "JSONCallback"+handler.hashCode();
        get(url+callbackName, callbackName, handler);
    }

    public static void get(String url, String callbackName,
                           JSONRequestHandler handler) {
        createCallbackFunction(handler, callbackName);
        addScript(url);
    }

    public static native void addScript(String url) /*-{
        var scr = document.createElement("script");
        scr.setAttribute("language", "JavaScript");
        scr.setAttribute("src", url);
        document.getElementsByTagName("body")[0].appendChild(scr);
    }-*/;

    private native static void createCallbackFunction(
                                  JSONRequestHandler obj,
                                  String callbackName) /*-{
        tmpcallback = function(j) {
            obj.@com.kereki.client.JSONRequestHandler::
                onRequestComplete(
                   Lcom/google/gwt/core/client/JavaScriptObject;)(j);
        };
        eval( "window." + callbackName + "=tmpcallback" );
    }-*/;
}
```

Note that the last two methods are written in JavaScript instead of Java; the JavaScript code is written inside Java comments. The special @id... syntax inside the JavaScript is used for accessing Java methods and fields from JavaScript. This syntax is translated to the correct JavaScript by GWT when the application is compiled. See the GWT documentation for more information.

Listing 3. Source Code for the JSONRequestHandler Class

```
package com.kereki.client;
import com.google.gwt.core.client.JavaScriptObject;
public interface JSONRequestHandler {
    public void onRequestComplete(JavaScriptObject json);
}
```

You can find the code for this listing and the previous one at **www.gwtsite.com/code/webservices**.

Advertiser Index

Listing 4. Source Code for the Main Program

```
package com.kereki.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.core.client.JavaScriptObject;
import com.google.gwt.user.client.ui.*;
import com.google.gwt.json.client.*;
import com.google.gwt.http.client.URL;
import com.kereki.client.JSONRequest;
import com.kereki.client.JSONRequestHandler;

public class example implements EntryPoint {
    public void onModuleLoad() {
        final TextBox tbSearchFor = new TextBox();

        final TextArea taJsonResult = new TextArea();
        taJsonResult.setCharacterWidth(80);
        taJsonResult.setVisibleLines(20);

        final HorizontalPanel hp1 = new HorizontalPanel();

        Button bGetNews = new Button("Get news!",
            new ClickListener() {
                public void onClick(Widget sender) {
                    JSONRequest.get(
                        "http://search.yahooapis.com/"+
                        "NewsSearchService/V1/newsSearch?"+
                        "appid=YahooDemo&query="+
                        URL.encode(tbSearchFor.getText())+
                        "&results=2&language=en"+
                        "&output=json&callback=",
                        new JSONRequestHandler() {
                            public void onRequestComplete(
                                    JavaScriptObject json) {
                                JSONObject jj= new JSONObject(json);
                                taJsonResult.setText(jj.toString());
                            };
                        }
                    );
                }
            });

        hp1.add(new Label("Search for:"));
        hp1.add(new HTML(" ",true));
        hp1.add(tbSearchFor);
        hp1.add(new HTML(" ",true));
        hp1.add(bGetNews);

        RootPanel.get().add(hp1);
        RootPanel.get().add(new HTML("<br>",true));
        RootPanel.get().add(taJsonResult);
    }
}
```

The code in Listing 4 shows access to a single service, but it would be easy to connect to several sources at once and produce a mashup of news.

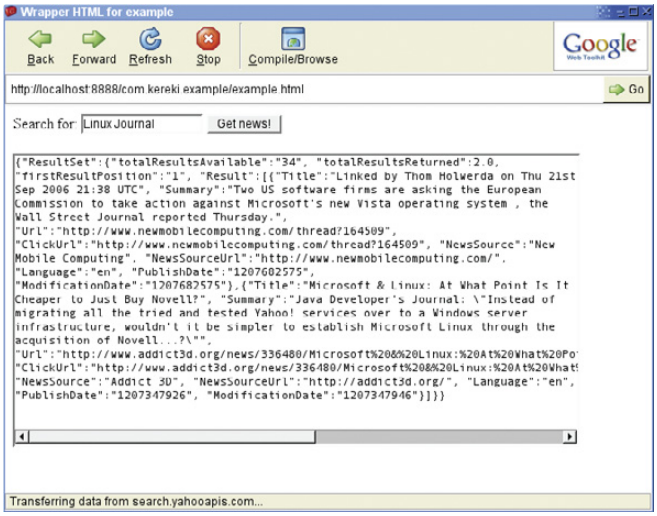


Figure 3. The Application, Running in Hosted Mode

Listing 5. Compiling the Code and Deploying the Files to Your Server

```
# cd ~/examplefiles/
# sh ./example-compile
Output will be written into ./www/com.kereki.example
Copying all files found on public pathCompilation succeeded
# sudo cp -R ./www/com.kereki.example /srv/www/htdocs/
```

In my case, with OpenSUSE, it's /srv/www/htdocs, but with other distributions, it could be /var/www/html (Listing 5). Users could use your application by navigating to http://127.0.0.1/com.kereki.example/example.html, but of course, you probably will select another path.

Conclusion

We have written a Web page without ever writing any HTML or JavaScript code. Moreover, we did our coding in a high-level language, Java, using a modern development environment, Eclipse, full of aids and debugging tools. Finally, our program looks quite different from classic Web pages. It does no full-screen refreshes, and the user experience will be more akin to that of a desktop program.

GWT is a very powerful tool, allowing you to apply current software engineering techniques to an area that is lacking good, solid development tools. Being able to apply Java, a high-level modern language, to solve both client and server problems, and being able to forget about browser quirks and incompatibilities, should be enough to make you want to give GWT a spin.■

Federico Kereki is a Uruguayan Systems Engineer, with more than 20 years' experience teaching at universities, doing development and consulting work, and writing articles and course material. He has been using Linux for many years now, having installed it at several different companies. He is particularly interested in the better security and performance of Linux boxes.

Resources

Google Web Toolkit Applications by Ryan Dewsbury, Prentice-Hall, 2008.

Google Web Toolkit for AJAX by Bruce Perry, PDF edition, O'Reilly, 2006.

Google Web Toolkit Java AJAX Programming by Prabhakar Chaganti, Packt Publishing, 2007.

Google Web Toolkit Solutions: Cool & Useful Stuff by David Geary and Rob Gordon, PDF edition, Prentice Hall, 2007.

Google Web Toolkit Solutions: More Cool & Useful Stuff by David Geary and Rob Gordon, Prentice Hall, 2007.

Google Web Toolkit—Taking the Pain out of AJAX by Ed Burnett, PDF edition, The Pragmatic Bookshelf, 2007.

GWT in Action: Easy AJAX with the Google Web Toolkit by Robert Hanson and Adam Tacy, Manning, 2007.

AJAX: a New Approach to Web Applications: **www.adaptivepath.com/ideas/essays/archives/000385.php**

AJAX: Getting Started: **developer.mozilla.org/en/docs/AJAX:Getting_Started**

AJAX Tutorial: **www.xul.fr/en-xml-ajax.html**

Apache 2.0 Open Source License: **code.google.com/webtoolkit/terms.html**

Eclipse: **www.eclipse.org**

Google Web Toolkit: **code.google.com/webtoolkit**

GWT4NB, a Plugin for GWT Work with NetBeans: **https://gwt4nb.dev.java.net**

Java SE (Standard Edition): **java.sun.com/javase**

Java Development Kit (JDK): **java.sun.com/javase/downloads/index.jsp**

JSON: **www.json.org**

JSON: the Fat-Free Alternative to XML: **www.json.org/xml.html**

NetBeans: **www.netbeans.org**

Same Origin Policy, from Wikipedia: **en.wikipedia.org/wiki/Same_origin_policy**

Web 2.0, from Wikipedia: **en.wikipedia.org/wiki/Web_2**

What Is Web 2.0?, by Tim O'Reilly: **www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html**

XML: **www.xml.org**

IMPROVED SCAFFOLDING FOR RUBY on RAILS

Using the **ActiveScaffold** plugin to improve the default Ruby on Rails layouts.

In the May 2007 issue of *Linux Journal*, I described my initial foray into the world of Ruby programming, combining Ruby with CGI and AJAX to produce a Web-based Ethernet Analyzer. Although I had fun putting that particular solution together, my real reason for getting to know Ruby was to allow me to work with Ruby on Rails, the highly regarded Web Application Framework (WAF).

I've looked at a number of WAFs available within the Perl and Python spaces. Way back in the March 2005 issue of *Linux Journal*, I described Maypole, one of Perl's first WAFs. Since then, I've explored Catalyst (a Maypole fork), Jifty and Gantry. Despite my extensive use of and acknowledged fondness for Perl, Rails had caught my eye, and it was an itch I just had to scratch.

PAUL BARRY

Learn Ruby before Learning Rails

I have one piece of advice for anyone hoping to work with Rails in any meaningful way: get to know Ruby first. I initially had terrible trouble getting my head around what Rails was doing due to my inexperience with Ruby. When I changed my approach and set aside Rails in order to learn Ruby properly, my second pass at Rails made more sense. It was also more productive.

Is Rails Meant to Be This Ugly?

There is little doubt that Rails is a great WAF, worthy of all the praise continually heaped upon it. However, when you first start working with Rails, the default Web pages generated by the framework are anything but impressive. In fact, they are downright ugly, which can be a bit of a disappointment, especially if all you need is a quick Web application mockup. Granted, these default layouts are designed to be replaced by something nicer: professionally designed CSS Web pages. And, to be fair, the Rails folks do go to great lengths to stress this fact. However, if you are in a hurry, stopping to design some customer-friendly Web pages is a drag. What's needed is nice, modern CSS styling for the quick-and-dirty, in-a-hurry types like me. That's where ActiveScaffold comes in.

ActiveScaffold: a Rails Sweetener

ActiveScaffold is built on top of the standard Rails environment and is a plugin that, in the words of the project's Web site, "provides you with a wealth of dynamically created goodness". What this goodness means to Rails developers is that ActiveScaffold provides a nice set of CSS pages and methods for interacting with your database tables. ActiveScaffold initially manages to do this, somewhat remarkably, with only a single, trivial code change to an existing Rails application.

In this article, I redevelop the Web-based soccer club database application that I created with Maypole back in 2005, this time using Rails with ActiveScaffold as the development platform. To add a slight twist to the proceedings, I use PostgreSQL as my database, as I've decided to give PostgreSQL a go having read Reuven Lerner's excellent series of articles comparing PostgreSQL to MySQL (see the April, May and June 2007 issues of *LJ*).

Preparing PostgreSQL

If you don't have PostgreSQL installed (and you are using Ubuntu or some other Debian-based distro), installation is straightforward:

```
sudo apt-get install postgresql
```

If your GNU/Linux distribution does not support apt, use your package manager to download and install PostgreSQL. With PostgreSQL running, become the postgres user on your system and create a new soccer_manager user:

```
sudo su - postgres
createuser -U postgres soccer_manager
```

Be sure to answer n (for no) to each of the questions posed by the createuser program, as the soccer_manager needs to be restricted to working solely within the soccer database (which we'll create in just a moment). Selecting n deliberately restricts the privileges awarded at this stage. Next, create a database, called soccer_development:

```
createdb -U postgres soccer_development
```

With the database and user created, enter the PostgreSQL interactive terminal (psql), and give the soccer_manager a password as well as user privileges to use the soccer_development database:

```
psql
postgres=# alter user soccer_manager with
postgres-# password 'soccer_manager_password';
postgres=# grant all privileges on database
postgres-# soccer_development to soccer_manager;
postgres=# \q
```

Note the use of the quit command, \q, which exits psql. At this point, we are done working directly with PostgreSQL. We could log in to psql as the soccer_manager user and start to create tables within the database using standard SQL, but we'll get Rails to handle these details for us (more on this in a little while).

Configuring Rails for PostgreSQL

I'm assuming you already have Ruby installed on your GNU/Linux system. If this is not the case, either install it from source from the Ruby Web site (see Resources) or install the Ruby package from your distribution's package manager (the ruby-full package on Ubuntu should include all you need). To install and use Rails, the RubyGems Package Manager needs to be installed into your Ruby environment. If RubyGems is not available within your distribution's package manager, pop on over to the RubyGems download page on RubyForge (see Resources), select the version of RubyGems that best matches your environment, and download the associated file. Installation is straightforward (note that the version you are working with may differ from that shown here):

```
tar zxvf rubygems-1.3.0.tgz
cd rubygems-1.3.0
sudo ruby setup.rb
```

If you are using Ubuntu (or one of its cousins), install the RubyGems package using apt:

```
sudo apt-get install rubygems
```

With RubyGems installed, you now can install Rails:

```
sudo gem install rails
```

Be sure to install all the suggested dependencies when prompted. This step takes a little while to complete, but it is a testament to the simplicity of Rails that you are ready to go once this command completes. One of the problems I've experienced with Perl-based WAFs is that installation can be a nightmare, especially when different versions of various CPAN modules throw up compatibility and dependency errors.

Thankfully, there's no such maddeningly frustrating problems with Rails!

I did have one small problem with Rails on Ubuntu, which relates to the installation of the rails command in /usr/bin/, in that it wasn't there. Ubuntu expects you to install Rails using apt-get, but as I wanted the latest-and-greatest Rails, I went with the RubyGems installation method. To fix this small problem, create a link to the rails command, as follows:

```
sudo ln -s /var/lib/gems/1.8/bin/rails /usr/bin/rails
```

As we are using PostgreSQL as our database, we need to download and install the PostgreSQL Ruby gem. This, too, is straightforward:

```
sudo gem install postgres
```

If this causes an error, make sure the development libraries for Ruby are installed (called ruby1.8-dev on Ubuntu), as well as those for PostgreSQL (called libpq-dev). If compile-time errors still result (due to header files not being found, for instance), use this command instead (which should be entered on a single line):

```
POSTGRES_INCLUDE=/usr/include/postgresql   \
sudo gem install postgres
```

At this point, Ruby, PostgreSQL, the PostgreSQL gem and Rails are installed and ready for action.

Creating a New Rails Application

In a directory of your choosing, type the following command:

```
rails soccer_club --database=postgresql
```

This command creates a new Rails application called soccer_club, resulting in a long list of messages from Rails, and creates a new directory called soccer_club.

Preparing the Database Connection

Let's add some database tables to our application. Begin by first changing into the newly created soccer_club directory.

We could create the necessary tables using a series of SQL CREATE TABLE statements, patiently entering them into PostgreSQL's psql command-line tool. However, Rails provides a technology called Database Migrations that allows you to manipulate your database tables without directly using SQL. Migrations operate at a higher level, shielding the Web developer from the underlying SQL dialect. Before we create a Migration, let's tell our Rails application which database to use and provide a user name/password combination.

Edit the config/database.yml file associated with your Rails application, and change the development section to look like this (note that some default values have been *suggested* by Rails, but for our application, those values need to change):

```
development:
  adapter: postgresql
  encoding: unicode
  database: soccer_development
  username: soccer_manager
  password: soccer_manager_password
```

On my Ubuntu system, PostgreSQL is configured to expect connections from a user name equal to the user ID of the currently logged-in user. This is called IDENT Authentication. What this means is that to access the soccer_development database with user ID soccer_manager, we need to be logged in to GNU/Linux as soccer_manager. That's not what we want (and it's not what Rails wants either), so we need to make a quick change to the bottom of the appropriate PostgreSQL configuration file (/etc/postgresql/8.3/main/pg_hba.conf), commenting out the `ident sameuser` line and adding a password line, as follows:

```
# "local" is for Unix domain socket connections only
local       all      all              password
# local     all      postgres        ident sameuser
```

After that edit, it's necessary to stop/start PostgreSQL to apply the change:

```
sudo /etc/init.d/postgresql-8.3 stop
sudo /etc/init.d/postgresql-8.3 start
```

To check that all is well with the Rails connection to the database, type the following within the top-level directory of your Rails application:

```
rake db:migrate
```

A single line of output results (in /home/barryp/rails/soccer_club on my system), which is Rails' way of telling us that everything is okay with the database connection. Any other message may indicate an error. If it is not immediately clear what the problem is (assuming, of course, that you have one), try appending `--trace` to the end of the above rake command.

Creating the Database Tables with Rails

Rails can help with the creation of our database tables, and we need three: one to hold information on our soccer players, another for squad data and another to maintain medical conditions. For the sake of simplicity, let's assume that each player belongs to one squad and can have a single medical condition (or none at all). Let's tell Rails about the tables:

```
ruby script/generate model player
ruby script/generate model squad
ruby script/generate model condition
```

Models in Rails let us talk to our data from our Web application. Each of the above commands produces eight lines of output while Rails does its thing. Note that each contains a file generated in the db/migrate directory. These are our database migrations. At this point, things get less SQL-centric and more Rails-like, as Rails provides a database-independent way to define our tables. To see this in action, edit the db/migrate/xxxxxxxxx_create_players.rb file (where xxxxxxxxx is

Get Further with PHP

March 4-6, 2009
Montreal, Canada

CMS framework
database open
source 2.0 AJAX
RIA mobile RegEx
scalability security
PHP optimization
web services OOP
design patterns

Register online and get a 100$ discount
http://conf.phpquebec.com/lj

Registration ends
February 28th

PHP Quebec **2009 Conference**
March 4th to 6th in Montreal

Early Bird Prices Currently Available

a unique date/time string generated by Rails), changing the self.up method to look like this:

```
def self.up
  create_table :players do |t|
    t.integer     :squad_id, :condition_id
    t.string      :name, :address, :contact_tel_no
    t.date        :date_of_birth
    t.timestamps
  end
end
```

This is the high-level Rails way of telling your database to create a table. Each column in the table gets a unique name and a data type. Note that in addition to the columns you might expect each player to have (name, address and so on), we add in two integer columns that will link to the squad and condition tables. What's cool about using migrations is that it does not matter which database you are using, Rails generates the correct database-specific SQL statements as required and when needed. Let's define the other two tables. Edit db/migrate/xxxxxxxxxx_create_squads.rb, changing the self.up method as follows:

```
def self.up
  create_table :squads do |t|
    t.string          :name
    t.timestamps
  end
end
```

And, finally, change db/migrate/xxxxxxxxxx_create_conditions.rb to have a self.up method that looks like this:

```
def self.up
  create_table :conditions do |t|
    t.string          :name
    t.timestamps
  end
end
```

Now for the fun part, type the following at the command-prompt:

```
rake db:migrate
```

Output similar to the following should scroll by on screen:

```
(in /home/barryp/rails/soccer_club)
== CreatePlayers: migrating ======================
-- create_table(:players)
   -> 0.1916s
== CreatePlayers: migrated (0.1918s) ============

== CreateConditions: migrating ===================
-- create_table(:conditions)
   -> 0.0183s
== CreateConditions: migrated (0.0185s) =========

== CreateSquads: migrating =======================
-- create_table(:squads)
   -> 0.0309s
== CreateSquads: migrated (0.0311s) =============
```

What's happened is that Rails has connected to the back-end database and created the three required tables. Note that there's no programmer-written SQL code in sight! Rails handles all the down-and-dirty SQL details. For those readers who don't believe me, log in to PostgreSQL as soccer_manager and bask in the glory of the table schema that Rails has created for you.

The Default Rails Layouts

At this point, it would be normal to use Rails to generate some scaffolding code, then reach for a CSS reference to pretty up the whole thing. This is doable, but it takes time. For now, let's use Rails to generate empty controllers with these three commands:

```
ruby script/generate controller player
ruby script/generate controller squad
ruby script/generate controller condition
```

Each of these commands produces seven lines of output. Note that a Ruby file is generated in the app/controllers directory. These are source code files that will contain any business logic we want to add to our Rails application. We will do this in a little while. To complete the default Rails setup, we need to specify our table relationships. Edit app/models/player.rb to look like this:

```
class Player < ActiveRecord::Base
  belongs_to  :condition
  belongs_to  :squad
end
```

One Little Edit: ActiveScaffold Goodness

ActiveScaffold is written and maintained by a dedicated group of Rubyists who live at **activescaffold.com/team**. ActiveScaffold is a Rails plugin, and as such, gets installed into an existing Rails project, so let's do that first. From the top-level directory of your Rails application, type the following (which should be entered on a single line):

```
git clone git://github.com/activescaffold/active_scaffold.git \
          vendor/plugins/active_scaffold &&                   \
          rm -rf vendor/plugins/active_scaffold/.git
```

This command fetches ActiveScaffold and installs it into your Rails application. When this process completes, a new directory has been created within the vendor/plugins/ directory of your Rails application called activescaffold. For the plugin to work its magic, we need to create an application-level layout that will be used throughout our Rails application. Here's a bare-bones layout, which we need to create in the app/views/layouts directory and which is called application.rhtml:

```
<html>
<head>
```

```
 <title>Soccer Club Database System</title>
 <%= javascript_include_tag :defaults %>
 <%= active_scaffold_includes %>
</head>
<body>

<%= yield %>

</body>
</html>
```

This is a straightforward, essentially empty, HTML page. Take note of the code included within the <%= and %> tags. These tags allow us to execute Ruby code from within an HTML template. The first set of such tags adds a set of JavaScript routines to our page; the second pulls in the ActiveScaffold goodness, and the third executes the Ruby yield method. Any layouts that are created within our application (whether manually by us or dynamically by Rails or ActiveScaffold) will be wrapped in the application.rhtml layout, with their content replacing the invocation of yield as required. With the default layout created, we need to edit each of our existing controllers to switch on ActiveScaffold. Here's how the app/controllers/player_controller.rb file should appear after this edit:

```
class PlayerController < ApplicationController
  active_scaffold     :player
end
```

Add a similar line of code to the app/controllers/squad_controller.rb and app/controllers/condition_controller.rb files, then start your Rails application:

```
ruby script/server
```

Fire up your browser and load the http://localhost:3000/player page. Take a look at Figure 1, which shows the default ActiveScaffold player listing—it looks great. Note that ActiveScaffold has spotted the links between the three tables and pulled in the appropriate data values. Note also that I've added some sample data to my Web app. Unfortunately, the ordering of the columns leaves a little to be desired, and this is no more evident than when we view the default ActiveScaffold player form, as shown in Figure 2. This form displays the table columns in alphabetical order, which is not what we want. In addition, the subforms that provide access to the squad and medical condition data are cool, but what we want is a simple drop-down list for our application. Thankfully, adjusting ActiveScaffold's default behaviors is not difficult, as we shall see in a few moments.

Another problem (which you may have noticed if you've been following along) is that the date range associated with the date_of_birth value is very restrictive, using 1997 as the earliest start year. As all of our soccer players were born in the early 1990s, we need some way to adjust the start year for any entered dates. ActiveScaffold (together with Rails) can help here too.

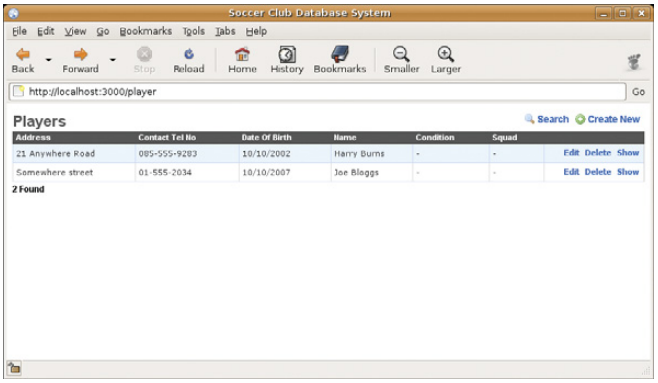


Figure 1. Default Player Listing as Generated by ActiveScaffold

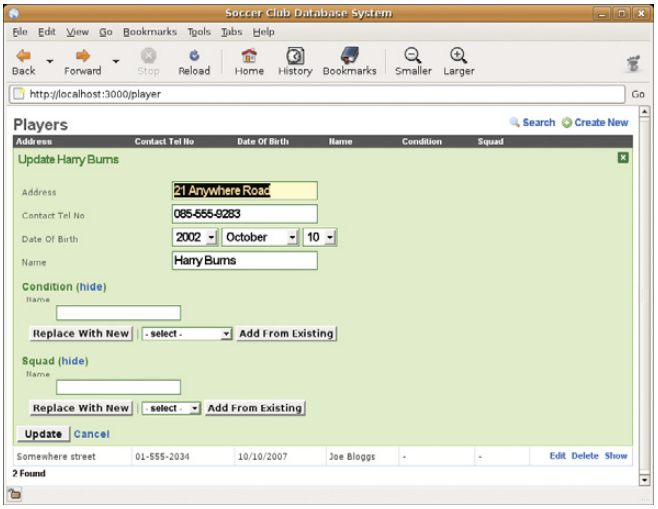


Figure 2. Default Player Data-Entry Form as Generated by ActiveScaffold

Refining ActiveScaffold's Behavior

Let's begin by fixing the order of our columns. Change the app/controllers/player_controller.rb file to look like this:

```
class PlayerController < ApplicationController
  active_scaffold :player  do |c|
    c.columns = [:name, :squad, :address,
                 :date_of_birth,
                 :contact_tel_no, :condition ]

    c.columns[:squad].ui_type = :select
    c.columns[:condition].ui_type = :select
  end
end
```

In this code, we provide a configuration code block to the activescaffold method where we specify the ordering of the columns, in addition to setting the ui_type associated with the squad and condition data to be :select. This fixes our ordering issue and sets the squad and condition selection mechanism to a standard drop-down list.

Sorting out the date problem requires the creation of

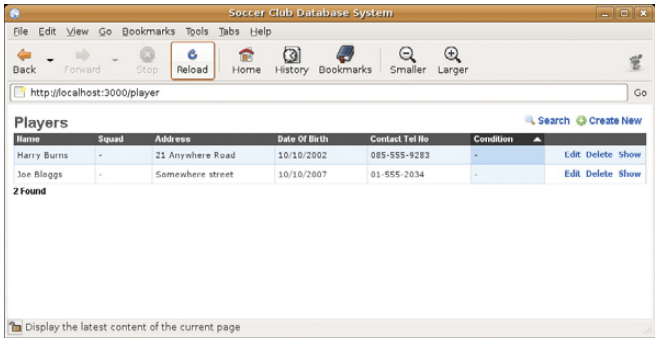


Figure 3. Improved Player Listing

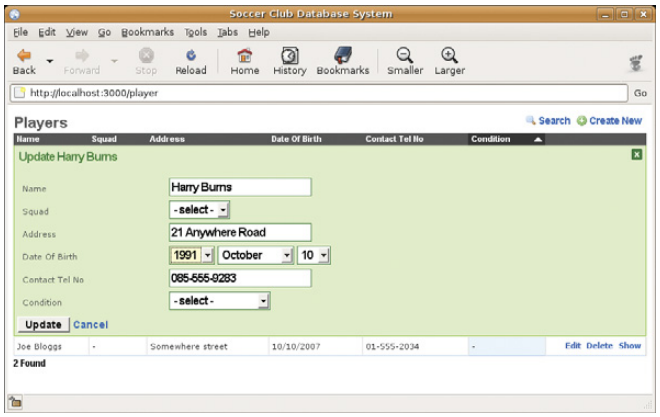


Figure 4. Improved Player Data-Entry Form

a Rails helper method for the players table. Edit the app/helpers/player_helper.rb file, and add the following code:

```
module PlayerHelper
  def date_of_birth_form_column(record, input_name)
    date_select :record, :date_of_birth,
                :name => input_name,
                :start_year => 1990
  end
end
```

The oddly named date_of_birth_form_column helper method calls the ActiveScaffold-supplied date_select method, which lets us adjust the earliest start date associated with our date_of_birth data. With these changes made, restart the Rails application and reload the browser window. Figure 3 shows the new-and-improved player listing, and Figure 4 shows the final version of our player data-entry form. As I'm sure you'll agree, both screens look the business. Take time to play around with the added functionality that ActiveScaffold has provided for free, including sort-ordering links on each of the column headings.

Learning More

To learn more about Rails, I highly recommend *Agile Web Development with Rails* by The Pragmatic Programmers (now in its second edition, with a third due soon), as well as O'Reilly Media's *Rails Cookbook*. To learn more about ActiveScaffold, check out the well-written documentation and code examples available on-line at the ActiveScaffold Web site (see Resources). As I hope this article demonstrates, it doesn't take much to turn an ugly, default Rails application into something you just might want to show off!■

Paul Barry (paul.barry@itcarlow.ie) lectures at the Institute of Technology, Carlow in Ireland. Find out more about the stuff he does at this Web site: glasnost.itcarlow.ie/~barryp.

Resources

Ruby on the Web: **www.ruby-lang.org**

The RubyGems RubyForge repository: **rubyforge.org/projects/rubygems**

ActiveScaffold Web Site: **activescaffold.com**

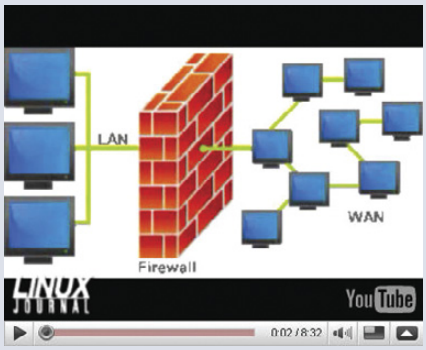
Rails Plugin Repository: **agilewebdevelopment.com/plugins**

"An Ajax-Enhanced Web-Based Ethernet Analyzer" by Paul Barry (*LJ*, May 2007): **www.linuxjournal.com/article/9614**

"A Database-Driven Web Application in 18 Lines of Code" by Paul Barry (*LJ*, March 2005): **www.linuxjournal.com/article/7937**

Reuven Lerner's Excellent Series of Articles Comparing PostgreSQL to MySQL (April, May and June 2007 issues of *LJ*): **www.linuxjournal.com/article/9571**, **www.linuxjournal.com/article/9618** and **www.linuxjournal.com/article/9649**

On the Web, Articles Talk!



Linux comes with a powerful firewall built in, although the interface can be a little intimidating. "Mastering IPtables" is a multipart tutorial on how to master basic and not-so-basic IPtables functionality and create the perfect firewall for your home network.

www.linuxjournal.com/video/mastering-iptables-part-i

Find Yourself in *Second Life* with Linux

Where open source, collaboration and gameplay unite.

BILL CHILDERS

Many Massively Multiplayer On-line Role-Playing Games (MMORPGs) exist on the Internet today. These games have their origins in the original Multi-User Dungeons (MUDs) available on early UNIX systems and bulletin boards, but they are scaled up to allow thousands of concurrent players and have very high-quality graphics. Some of the more well-known games in this family are *World of Warcraft* and *EverQuest*. However, there's another, fairly unique entry in this family of games—*Second Life*—that distinguishes itself from the rest for many reasons.

What is *Second Life*? Put simply, *Second Life* is a three-dimensional virtual world, developed by Linden Research, that lets its users explore, interact, socialize and even conduct business. Unlike some of the other games in the MMORPG family, *Second Life* is a "game about nothing". The game has no real goals and no stated ways to "win". Much like reality, when your character (or avatar, as it's known in *Second Life* parlance) is "born", you can go anywhere and do anything.

Perhaps the most unique thing about *Second Life* is not in its gameplay, nor in its technology. *Second Life* contains a set of tools that allows its players to generate their own objects in the game. These objects also can be scripted, using the Linden Scripting Language (LSL) or Mono. Once players create objects in the game, they own and retain the copyright to those items. As such, user-generated content can be given away or sold.

Second Life also has its own currency (the Linden Dollar, or $L). Players can buy Linden Dollars via a currency exchange run by Linden Labs, or they can earn money by generating their own content and selling it to other players within the game. This has allowed for a virtual economy to spring to life within the game, and some players have become prolific enough at creating content that selling virtual goods has allowed them to leave their real-world jobs behind. It's even possible to buy and sell virtual land (sim space) within the game, and many players invest heavily in land.

Second Life has been closely related with open-source projects since its inception. The game's client software (or viewer) contains many open-source components and has been built by Linden Labs to run on Windows XP, Mac OS X and Linux. In late 2007, Linden Labs announced it would open-source the viewer under the GPL. Linden Labs has made good on this promise, and as a result, several good alternative viewers have sprung up. There also is excellent documentation on how to build the viewer yourself, if you're inclined to fork the code and create your own viewer.

The server software Linden Labs uses is called the simulator (or sim for short) and still remains proprietary, although there are promises to open-source this component of the game as well. In lieu of Linden Labs releasing its sim software to the world, there has been a strong reverse-engineering effort in the community to create a functional equivalent of the sim server software. The OpenSimulator Project now has a functional alpha-quality server that's available under the BSD license. The OpenSimulator (OpenSim) code is written in C# and can run under .NET on Windows or Mono on Linux. This code is constantly under development, and new features are being added almost daily.

Each area in *Second Life* is called a region or sim (short for simulator). A sim is 65,536 square meters in size (16 virtual acres), and there are thousands of sims in the *Second Life* world. The sims are devoted to all manner of things, from science-fiction sims where you can fly your own X-Wing fighter, to medieval sims where you can be a Knight of the Round Table, to Roman sims where you can stage your own fall of the empire. If you can imagine it, it probably exists somewhere within *Second Life*.

Although *Second Life* has been dubbed a game, it has many uses besides the obvious entertainment value. One of the most interesting purposes for *Second Life* has been its adoption by many universities as a way to augment distance learning. As of this writing, more than 100 sims are devoted to educational purposes. Universities with a presence in *Second Life* include Princeton University, Rice University, Stanford University and the National University of Singapore.

Religious organizations also have found *Second Life* to be a useful vehicle for spreading their message. And, I've heard stories where *Second Life* has been a form of therapy for people with disabilities and other challenges—although those people have challenges in real life, in *Second Life*, they are free of any limiting factors.

Second Life also provides a rich medium for the arts—not only can you find live music and concerts (similar to Webcasts) for any music genre imaginable, but there also are new forms of art unique to *Second Life*. There is machinima, or movies that are made using nothing but *Second Life* avatars, build tools and cameras. Creating sims also can be considered an art form, and some breathtakingly beautiful and unique places have been created within the game. *Second Life* avatars even can be considered art forms themselves!

Many businesses also have taken up a presence in *Second Life*. IBM and Sun Microsystems maintain a presence in the *Second Life* metaverse, as do other firms, such as telecommunications companies Vodaphone and Swisscom. IBM also maintains many research sims, and it is a contributor to the

OpenSimulator Project. The Reuters news agency also maintains a *Second Life* presence and has positioned itself as the news media for the virtual world, even going so far as to have dedicated reporters working inside *Second Life*.

Second Life can act as a fairly full-featured teleconferencing suite. The client includes a voice chat component (which does work under Linux), so having a meeting in the *Second Life* metaverse is really quite easy.

Several Linux User Groups (LUGs) are active and meet on a semi-regular basis inside the game, so if you're living in a rural location and it's difficult for you to attend your local LUG, *Second Life* may be a viable alternative for you to meet other Linux users and discuss your favorite operating system.

In addition to the voice chat, *Second Life* also has group and private instant-messaging (IM) functionality, though this does not leave the *Second Life* world and extend into the real world...yet. Linden Labs has mentioned possible XMPP (Jabber) integration and a gateway into *Second Life* for some time, as well as the possibility of a "light" client being created to allow for text-based chatting without all the graphical effects. Third-party viewers have this functionality already, such as the Web-based AjaxLife.

Getting started in *Second Life* is simple. In a Web browser, go to the *Second Life* "Create an Avatar" Web site (see Resources), and fill out the necessary form. The form directs you to download the client for your platform and provides instructions on how to install it.

If you use Linux as your operating system while running *Second Life*, make sure you're using hardware that meets or exceeds the recommended hardware list as published by Linden Labs. *Second Life* is *very* graphics-intensive and will tax your hardware to its limit. Although you can get around using a system with an integrated graphics chipset (I have used an X60 with the Intel 945 chipset before), it's not a pleasant experience to explore a virtual world at a two-frame-per-second refresh rate. I recommend using an NVIDIA or ATI graphics card with a minimum of 64MB of RAM that also has decent Linux drivers. If there is an optimal way to configure your card (nvidia-glx or the NVIDIA binary drivers), I recommend you do so. It can mean the difference between smoothly navigating the virtual world or watching it flip by one frame at a time.

Once you get in the game, you land on an Orientation Island. This island has mini-tutorials on basic *Second Life* skills, such as controlling your avatar and communicating with other residents. Take the few minutes necessary to learn the ways to move and interact within *Second Life*—it's time well spent that will help alleviate frustration later. Once you've mastered the basics, you can click on an object on the island that teleports your avatar to the mainland where you can begin exploring.

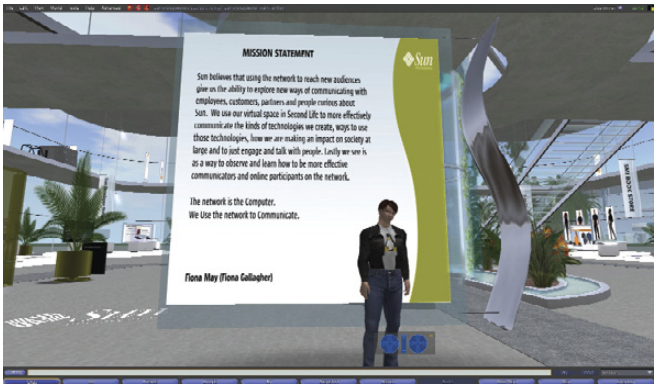


Figure 1. Visiting Sun Microsystems

Once you've left the safety of Orientation Island, you're on

your own. From here, it really is a case of "where do you want to go today?" *Second Life* includes a robust search engine to help you identify places to go and things to do that meet your interests. I took the time to build a quick sample tour of some of the things within *Second Life*. This is only a small slice of what can be done inside the virtual world of *Second Life*. There even are games within the game, each with its own rules and miniature communities.

To get a taste for what is in *Second Life*, let's take a quick tour. The Resources section of this article includes links to these locations (otherwise known as SLURLs), so you can visit them yourself. Let's begin our virtual tour at the Sun Microsystems pavilion within *Second Life*.

Many large corporations like Sun and IBM have established presences within Second Life for sales and support personnel to reach out to the avatar community. Frequently, these companies give away free virtual goods here—at Sun's pavilion, you can get a free virtual 24" flat-screen monitor and a free virtual Ultra 40 workstation. These freebies don't do much (they're little more than static props), but if you're building a virtual workspace or need a prop for some reason, these things can come in handy. Figure 1 shows my avatar standing before the Sun Mission Statement. Do you like my fancy Linux T-shirt?

From here, let's teleport over to the Stanford University Library simulation. This sim contains resources and exhibits for the various Stanford University libraries, and the exhibits seem to change on a semi-regular basis. There are also several news boards here that rotate through Stanford-specific information pertaining to events within the sim, as well as events on the real Stanford campus. Wander this virtual campus enough, and you may bump into virtual students and virtual professors!

Our next stop is an example of the unique forms of art that can only exist within the *Second Life* world. I had spent some time looking for a good example of the *Second Life*-specific art, and I finally found it on the Summer sim. Summer is a faerie-themed sim that's made to appear magical

Figure 2. Stanford University Libraries

Figure 3. Summer Seale—Avatar as Art

and uses one-of-a-kind glow effect and particle displays to achieve this look. From huge mushrooms to glowing will-o'-the-wisps to a miniature Stonehenge, this sim has little eye-catching details placed perfectly throughout its 16 virtual

Alternative Viewers

Since Linden Labs released the viewer source code under the GPL, quite a few forks of the code have emerged. There is a page on the *Second Life* wiki that tracks these forks (see Resources), but here are a few of the most popular ones:

- Cool SL Viewer: Henri Beauchamp has taken many of the outstanding bugfixes in the JIRA bug-tracking system for *Second Life* and applies those fixes to the viewer before Linden Labs does. The result is an incredibly full-featured viewer that actually may perform better than the official viewer. Available for Linux, Windows and Mac.
- Dale's SL Viewer: Dale Glass has added some features to the viewer, such as a radar feature that tells you who's nearby, as well as support for 3-D stereoscopic goggles. Available for Linux and Windows.
- Onrez Viewer: a custom viewer created for a television show (*CSI: New York*) and *Second Life* tie-in. Windows only.
- AjaxLife: Katherine Berry is a talented programmer who's coded a text-only Web interface for *Second Life* called AjaxLife. Cross-platform support included—it even runs on the iPhone!

Figure 4. Captain of the Ship

acres. While wandering the sim, I was lucky enough to meet its owner, Summer Seale, and she spent a moment to show me one of her particle-based art forms, something she calls Chaos Theory. Her avatar is perfect example of the avatar as an art form—she looks like she'd stepped out of a fantasy movie. Figure 3 shows her in front of her fireworks-like Chaos Theory.

Our final stop on this slice of *Second Life* is the Sci-Fi Works store on Thunderbird Island. This place sells amazingly accurate reproductions of science-fiction spaceships and other sci-fi props. If you want to get an X-Wing fighter and hold a dogfight with a friend who's flying a TIE fighter, this is the place. Or if you're a *Star Trek* fan and want to pilot your own *Enterprise* or beam down to an alien world, that's an option too.

I've touched on only a few of the things *Second Life* can offer. Whether your interests lie in creating and selling your own content inside the game, in using the platform as a collaboration or educational tool, in creating new and singular art forms, or in actually using the platform for entertainment purposes, the *Second Life* virtual world has a lot to see and do.

In part II of this article, I'll detail how you can run an OpenSimulator instance on your own hardware. Imagine...a private collaboration and teleconferencing service for your business or simply a sandbox for *Second Life* development to occur. Stay tuned!

Resources

Second Life Home Page: **www.secondlife.com**

Create an Avatar: **secondlife.com/join**

Second Life Wiki: **wiki.secondlife.com**

OpenSimulator: **opensimulator.org/wiki/Main_Page**

Alternate *Second Life* Viewers: **wiki.secondlife.com/wiki/Alternate_viewers**

AjaxLife: **ajaxlife.net**

Compiling the *Second Life* Viewer for Linux: **wiki.secondlife.com/wiki/Compiling_the_viewer_(Linux)**

Sun Microsystems (SLURL): **slurl.com/secondlife/Sun%20Microsystems%201/128/86/71**

Stanford University (SLURL): **slurl.com/secondlife/Stanford%20University%20Libraries/159/227/33**

Summer (SLURL): **slurl.com/secondlife/Summer/106/35/22**

Sci-Fi Works (SLURL): **slurl.com/secondlife/Thunderbird%20Island/25/237/312**

Acknowledgements

Thanks to *Second Life* residents Neural Blankes and Summer Seale for their help with this article.■

Bill Childers is an IT Manager in Silicon Valley, where he lives with his wife and two children. He enjoys Linux far too much and probably should get more sun from time to time. If he ever finds the time to make it into *Second Life*, he goes by Bill Deere.

EOF

Net Development

The platform is the Net. Not just the Web. That's why we need to open it.

DOC SEARLS

It's important to remember that the Web began as a project. As Tim Berners-Lee explained (in an August 1991 post to alt.hypertext, **groups.google.com/group/alt.hypertext/msg/395f282a67a1916c**), "The WWW project merges the techniques of information retrieval and hypertext to make an easy but powerful global information system."

Nearly two decades later, the Web has done exactly that—and much more. While still the province of search engines and browsers, it also includes a collection of utilities we call the cloud, backed by massive storage and compute capacities residing in the racks of Amazon and Google. That's in addition to countless Web services, applications and other graces of development work (such as we cover in the preceding pages of this month's *Linux Journal*).

Yet no matter how large and encompassing the Web becomes, the Net remains the broader platform, the more encompassing environment. Everywhere society has digital foundations, the Net is there to make the connections. Today those connections span the whole world. So why don't we hear more about Net Development?

Although there are plenty of Internet protocols and applications outside the Web (IM, file syncing and sharing, and e-mail all jump to mind), we tend not to think of the Net as a platform. Perhaps that's because the Net's protocol suite is about transport rather than presentation or application. It doesn't care what datalinks (Ethernet, DSL, WDM, MoCA) or what physical or wireless media (copper, fiber, Wi-Fi, 3G, WiMAX) are used. It just makes a best effort over what's available.

And, that's the gating factor: what's available.

Today, most of us get on the Net through a phone or cable company that sells Net access as the third act in what they call a triple play. The first two acts are telephony and television. The Comcast Triple Play, for example, is pitched as "The best in TV, phone and Internet—three great services. One low bill. Hey, life just got a little easier." This positions the Net as just another "service", on par with television and telephony. Never mind that the Net can encompass both.

And they don't give us the whole Net. They cripple it with asymmetrical provisioning (even fiber deals default to higher downstream than upstream bitrates), blocked ports and lack of fixed IP addresses. If we want more, we have to move up to a "business" tier that begins with lower data rates and much higher prices—a shakedown racket that persists from the days when Ma Bell and national PPTs ruled the Earth.

The Net most of us know best is one where the Web is a wide-open platform for development, while the Net it runs on is "delivered" as a data spigot. Back when the carriers first realized that they were now ISPs, the Internet service they thought they'd be providing was biased by what they knew best and expected people would want: entertainment on the TV model. That usage materialized, but so did countless others. The carriers continue to miss a lesson of Web development that has thrived in spite of carriers' asymmetrical biases: that open platforms and without commercial biases support an infinitude of business. The Web is generative. (As Jonathan Zittrain puts it in *The Future of the Internet and How to Stop It*—for more, see "A Tale of Two Futures", the EOF from July 2008.) They don't yet see how selling the Net as just one (crippled) "play" forecloses an infinitude of other plays.

But the tide is starting to turn. In November 2008, I attended a "brainstorm" conference in London, put on by the Telco 2.0 Initiative, the mission of which is to "catalyze change in the Telecoms-Media-Technology sector". Every speaker and panelist inveighed, one way or another, against "triple play" and every other doomed-monopoly business model. Instead, they expanded on this advice in the Telco 2.0 Manifesto (**www.telco2.net/manifesto**):

> New value lies in addressing the friction that exists in everyday interactions between businesses and consumers, and governments and citizens. Typical examples include: authenticating users, market research, targeting promotions, distributing goods and content, collecting payments and providing customer care....
>
> Telcos collectively have assets that can address this situation: real-time user data, secure distribution networks, sophisticated payment processing capabilities, trusted brands, a near universal subscriber base, as well as core voice and messaging products.

Problem is, this still positions carriers as intermediaries between businesses and consumers. It ignores the enormous reservoir of production capacity on the "consumer" side, both by individual users and by developers—two parties who have been dancing away on the Web's wide-open floor.

The big money for carriers isn't just going to be in B2B and B2C. It will be in supporting all kinds of new activities made possible by a wide-open Net: one no longer biased toward single uses and no longer priced to discourage productive involvement by individuals and small businesses.

For that to happen, we need developers to step up with ideas that are Net-based and not just Web-based—ideas that help carriers leverage benefits of incumbency other than old monopoly businesses.

There are clues in handhelds. The best "smartphones" are computing devices on which voice is just one among thousands of applications. (See "Smarter Than Phones" in this issue's UpFront section.) More important, the user is in charge of more and more apps and what can be done with them.

Independence, autonomy and choice are going to be facts of connected life for every individual, sooner or later. So will unlimited data integration and production potential. The policies, preferences and terms of service that matter will be those asserted by individuals, not just those controlled by service providers and other sellers.

There are huge opportunities in figuring out ways to help individuals and businesses form new and symmetrical relationships—ones in which choice is maximized on both sides. But it won't happen until we make the Net as open as it was born to be. That's a huge project. And we've barely started on it.■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

GPU Computing

More GFLOPS, Fewer Watts!

High Density Tesla Clusters

WhisperStation™ –PSC with Tesla C1060 GPUs
Computing Clusters with Tesla S1070 - 4 GPU Servers

GPU Performance:

- 1 TFLOPS per GPU
- 4 GB DDR3 per GPU
- 102 GB/Sec Bandwidth

CUDA SDK

- Standard C language
- Standard numerical library- FFT and BLAS

36 GPUs + 36 CPUs
Computing Cluster in 24 U

FireStream™ Workstations & Servers

WhisperStation™ – FS with FireStream 9250 Stream Processors

AMD
Solution Provider
PLATINUM

AMD FireStream 9250

Stream Processor Performance

- 1 TFLOPS
- 1 GB DDR3
- Single Wide Card /150 Watts
- Double Precision Floating Point Hardware

Stream SDK

- Brook+ Open Source Compiler
- ACML and CAL Math Libraries

Microway®
Technology you can count on℠

508-746-7341
microway.com

GSA
GSA Schedule Contract Number: GS-35F-0431N