Shoulda | Openfire | Lazarus | AppArmor | tircd | Titanium | Qt

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community
SEPTEMBER 2009 | ISSUE 185 | www.linuxjournal.com

LAZARUS
a Powerful IDE for Cross-Platform Development

Qt
Making Apps Right on All Platforms

Build Rich Desktop Apps with TITANIUM

GOOGLE CHROME DEVELOPERS

Supercharging the Web with V8

Open-Source Compliance Best Practices

Looking at AppArmor in Ubuntu

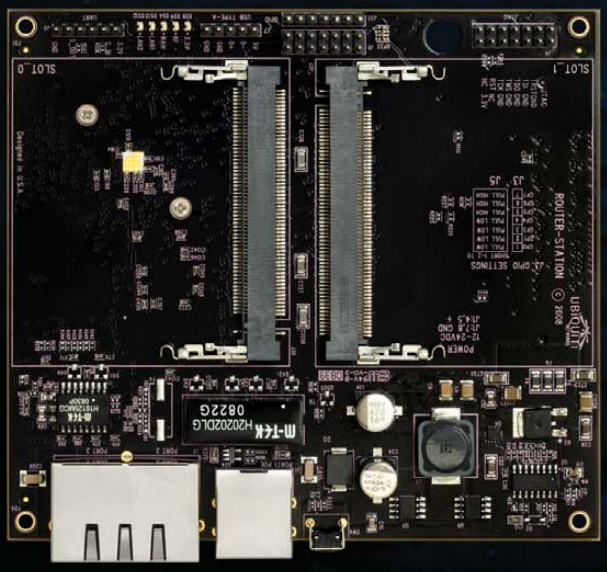
Use tircd to Connect to Twitter over IRC

SHOULDA MAKES TESTING EASIER FOR RAILS DEVELOPERS

$5.99US $5.99CAN
09
0 09281 03102 4

UBIQUITI NETWORKS

Embedded Wireless Dream Machines.

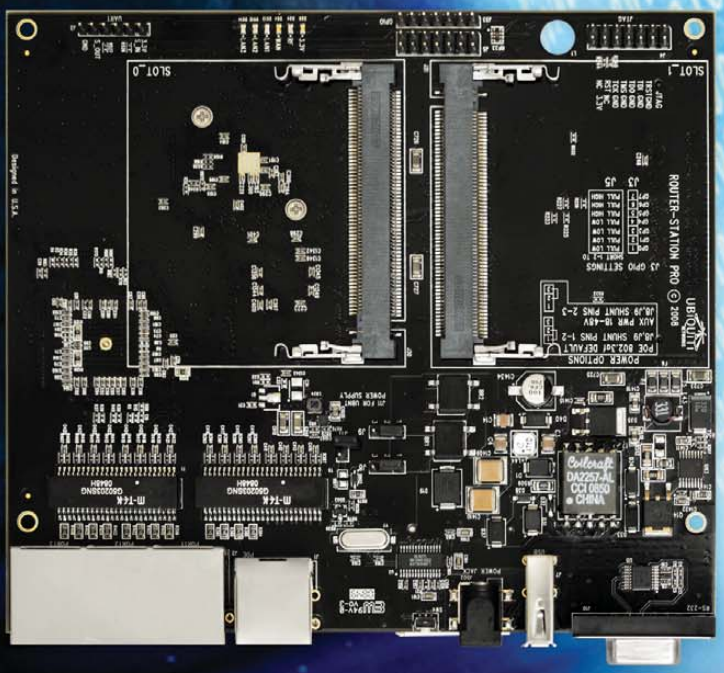


RouterStation

Featuring a fast 680MHz MIPS 24K CPU, 64MB RAM, and 16MB Flash; RouterStation provides a excellent horsepower for a variety of processor intensive multi-radio system applications.

Up to 3 mini-PCI radios, 3 10/100 ethernet interfaces, a 5A power supply for multiple hi-power card support, USB 2.0, and enhanced temperature operating performance and ethernet ESD protection for carrier applications.

MSRP $59



RouterStation Pro

In response to the outstanding demand for our initial RouterStation OEM platform, Ubiquiti Networks announces the RouterStation Pro. Breakthrough Price/Performance with a $79 USD MSRP.

Pro Version Enhancements:

- 48V 802.3af Power Over Ethernet
- 4-Port Gigabit Ethernet Switch
- 256MB RAM
- On Board SDIO Support
- On Board, USB 2.0, RS232/dB9, and DC power jacks

MSRP $79

Prices in USD. Ubiquiti Networks, Inc. Copyright © 2009 All Rights Reserved.

www.ubnt.com

Temperatures are rising – prices have dropped!

1&1 Summer Specials:

1&1® HOME PACKAGE

2 Domain Names Included
(Choose from .com, .net, .org, .biz and .info)

- 120 GB Web Space
- 1,200 GB Monthly Traffic
- 1,200 E-mail Accounts
- Easy-to-use Site Building Tool
- 1&1 Blog
- 1&1 Photo Gallery
- Credit to start advertising with major search engines like Google™.
- 24/7 Customer Support
- ... and much more!

1st year FREE!*

~~$4.99~~ per month

Offer valid through August 31st !

.us Domain Names $2.99 for the first year!*
More special offers are available online.
For details, visit www.1and1.com

Now accepting PayPal™

*Offers valid through August 31, 2009. 24 month minimum contract term and a setup fee of $4.99 apply with the Home Package offer. Other terms and conditions may apply. Private domain registration not available with .us domains. Visit www.1and1.com for full promotional offer details. Program and pricing specifications and availability 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.
© 2009 1&1 Internet, Inc. All rights reserved.

MEMBER OF united internet

Call **1-877-GO-1AND1**
Visit us now **www.1and1.com**

1&1®

CONTENTS

SEPTEMBER 2009
Issue 185

FEATURES

CROSS-PLATFORM DEVELOPMENT

48
GOOGLE CHROME: THE MAKING OF A CROSS-PLATFORM BROWSER
What does it take to make a cross-platform browser work well on three platforms?
James Gray

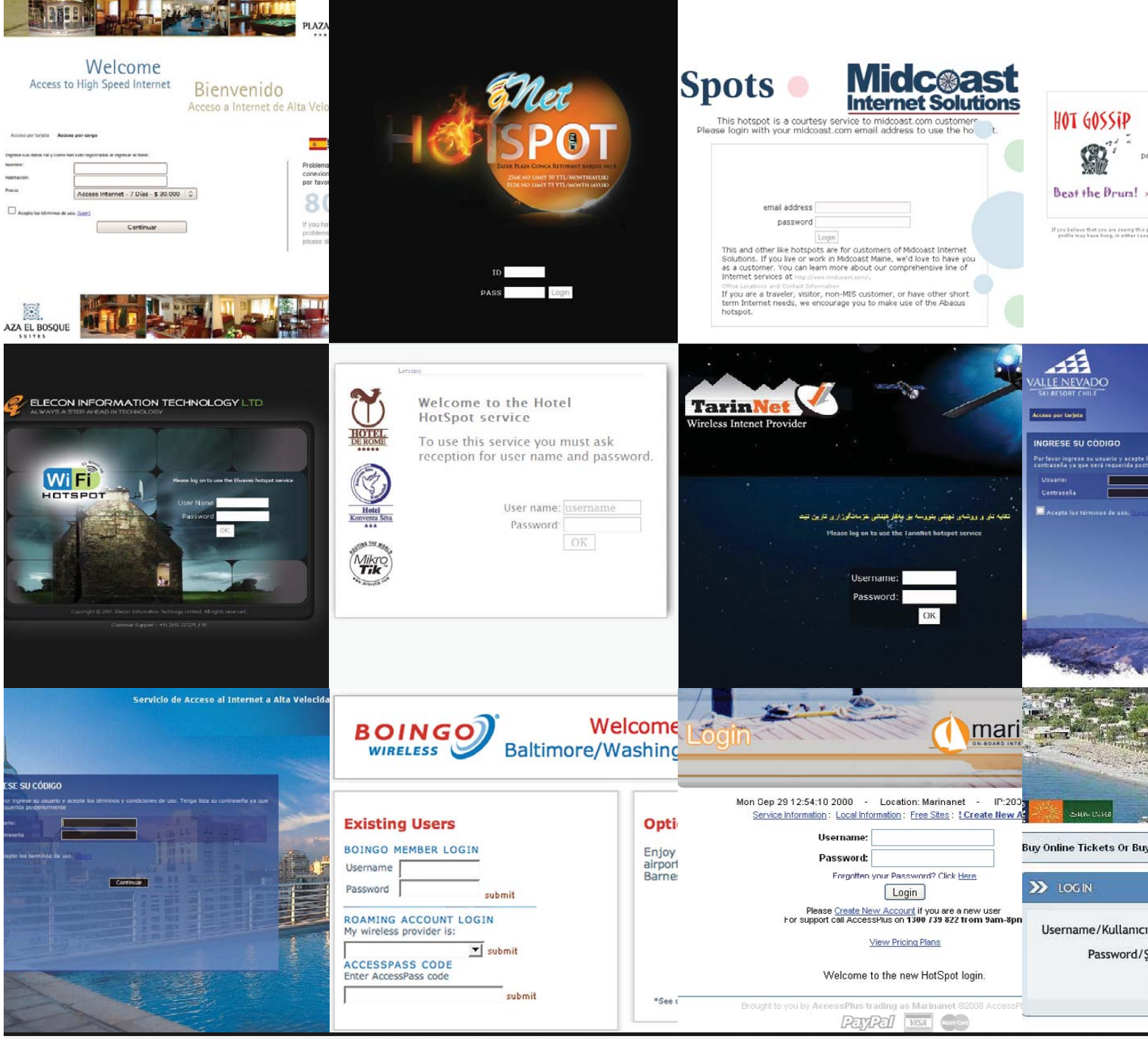
54
RICH CROSS-PLATFORM DESKTOP APPLICATIONS USING OPEN-SOURCE TITANIUM
Web developer, meet the desktop.
Mark Obcena

60
LAZARUS FOR CROSS-PLATFORM DEVELOPMENT
Pascal. Native code. Linux, Windows and Mac, oh my!
Mattias Gaertner

66
HOW TO BE CUTE ON ALL DESKTOPS WITH QT
It's not called Qt for nuttin.
Johan Thelin

ON THE COVER

- Google Chrome Developers—Supercharging the Web with V8, p. 48
- Lazarus—a Powerful IDE for Cross-Platform Development, p. 60
- Qt—Making Apps Right on All Platforms, p. 66
- Build Rich Desktop Apps with Titanium, p. 54
- Open-Source Compliance Best Practices, p. 72
- Looking at AppArmor in Ubuntu, p. 32
- Use tircd to Connect to Twitter over IRC, p. 40
- Shoulda Makes Testing Easier for Rails Developers, p. 18



MikroTik powered Hotspots around the world

MikroTik RouterOS powered hotspots are everywhere. From Internet cafes in desert towns of Africa, to futuristic airport lounges in the US and five star hotels in the Mediterranean. Mikrotik can power your systems too. Free evaluation installations are available in our download section.

www.mikrotik.com

CONTENTS SEPTEMBER 2009 Issue 185

COLUMNS

18 REUVEN M. LERNER'S AT THE FORGE
Testing Rails Applications with Shoulda

24 MARCEL GAGNÉ'S COOKING WITH LINUX
Cross at Your Platform?

30 DAVE TAYLOR'S WORK THE SHELL
Messing Around with ImageMagick

32 MICK BAUER'S PARANOID PENGUIN
AppArmor in Ubuntu 9

40 KYLE RANKIN'S HACK AND /
What Really IRCs Me: Twitter

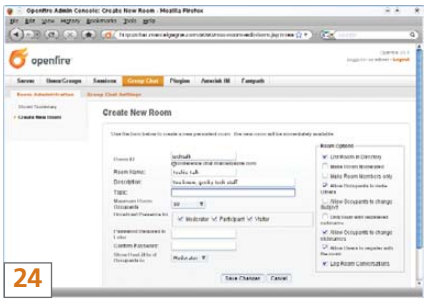
80 DOC SEARLS' EOF
Conferences: Pro & Un

INDEPTH

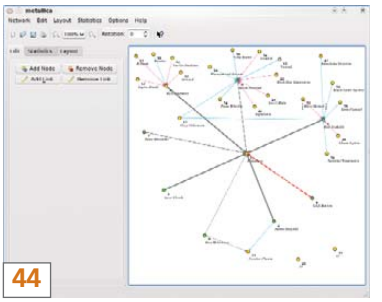
72 OPEN-SOURCE COMPLIANCE
Getting started guide and industry best practices.
Ibrahim Haddad

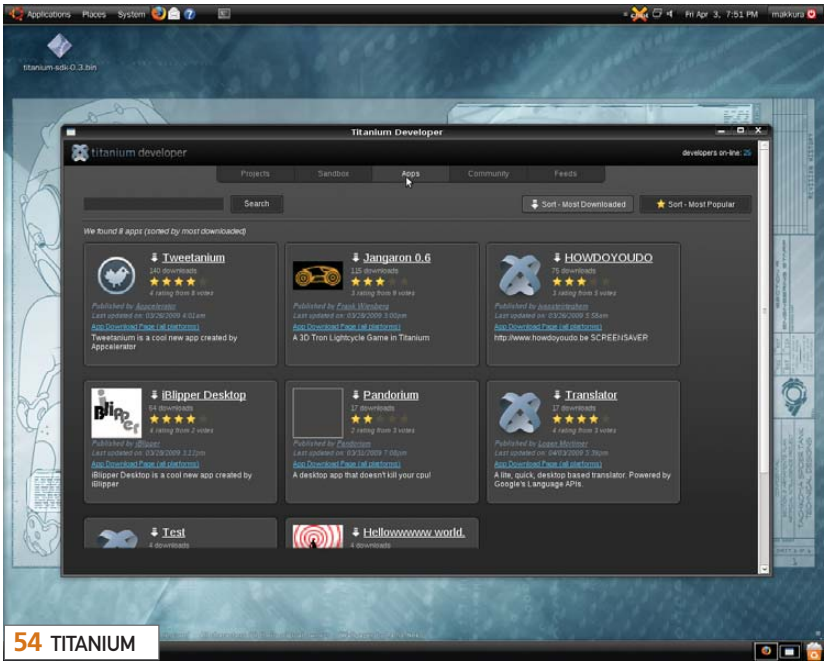
IN EVERY ISSUE

8 CURRENT_ISSUE.TAR.GZ
10 LETTERS
14 UPFRONT
42 NEW PRODUCTS
44 NEW PROJECTS
65 ADVERTISERS INDEX
78 MARKETPLACE



24


44



54 TITANIUM

Next Month

HACK THIS/PROGRAMMING HACKS

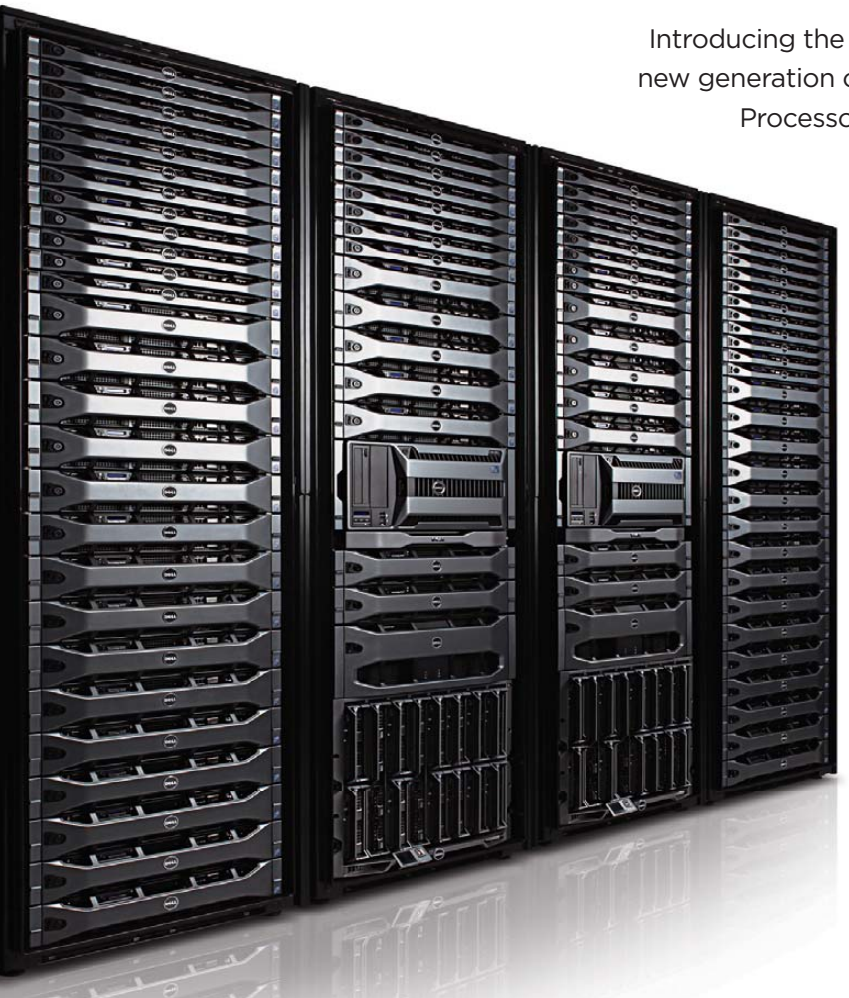
What happened to all the real Hackers? Well, they're standing by their mailboxes waiting for next month's issue.

In our upcoming Hack This/Programming Hacks issue, we've got just the kind of stuff Hackers like. Write your own OS, right down on the metal—virtual metal that is, using KVM. For the really brave, find out how to use Coreboot and get free of those proprietary BIOSes (or is that BIOSi?). Better yet, build your own CPU with an FPGA.

And, if you're goo goo over Google, find out how to run Android everywhere.

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

A MARVEL OF MODERN REENGINEERING.



Introducing the new Dell™ PowerEdge™ Server Solution, featuring a new generation of intelligent server processors with the Intel® Xeon® Processor 5500 Series. And the industry's best performance per watt. If you thought you knew Dell, think again.

Customized. Personalized. Recognized. Dell ranked #1 in server customer satisfaction.*

Future proof design. Dramatically reduce your power consumption and run more efficiently in your data center.

Reduced deployment time. The world's only server with instant-on embedded systems management. No media required.

SIMPLIFY IT. MINIMIZE RISC.
MIGRATE TO DELL POWEREDGE SERVERS AT DELL.COM/RISCMigration

*TBR x86-based Server Customer Satisfaction Study, Q4, February 18, 2009.

Intel, the Intel logo, Xeon and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

LINUX JOURNAL™
Since 1994: The Original Magazine of the Linux Community

Digital Edition Now Available!

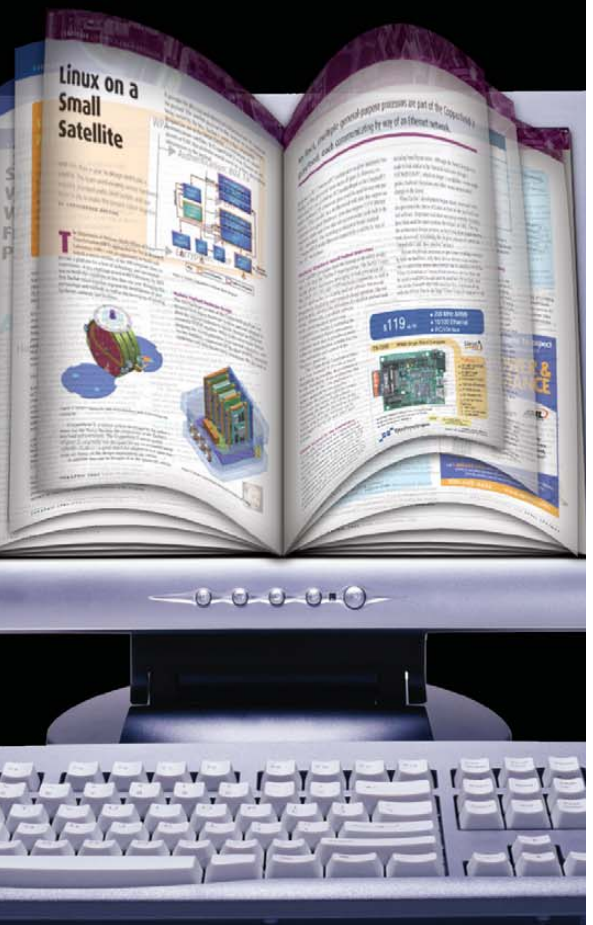
Read it first
Get the latest issue before it hits the newsstand

Keyword searchable
Find a topic or name in seconds

Paperless archives
Download to your computer for convenient offline reading

Same great magazine
Read each issue in high-quality PDF

Try a Sample Issue!
www.linuxjournal.com/digital



LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Associate Editor	Mitch Frazier mitch@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Chef Français	Marcel Gagné mggagne@salmar.com
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors
David A. Bandel • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti
Ludovic Marcotte • Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf

Proofreader Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Sales Manager	Joseph Krack joseph@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

Linux Journal **is published by, and is a registered trade name of, Belltown Media, Inc.**
PO Box 980985, Houston, TX 77098 USA

Reader Advisory Panel
Brad Abram Baillio • Nick Baronian • Hari Boukis • Caleb S. Cullen • Steve Case
Kalyana Krishna Chadalavada • Keir Davis • Adam M. Dutko • Michael Eager • Nick Faltys • Ken Firestone
Dennis Franklin Frey • Victor Gregorio • Kristian Erik • Hermansen • Philip Jacob • Jay Kruizenga
David A. Lane • Steve Marquez • Dave McAllister • Craig Oda • Rob Orsini • Jeffrey D. Parent
Wayne D. Powel • Shawn Powers • Mike Roberts • Draciron Smith • Chris D. Stark • Patrick Swartz

Editorial Advisory Board
Daniel Frye, Director, IBM Linux Technology Center
Jon "maddog" Hall, President, Linux International
Lawrence Lessig, Professor of Law, Stanford University
Ransom Love, Director of Strategic Relationships, Family and Church History Department,
Church of Jesus Christ of Latter-day Saints
Sam Ockman
Bruce Perens
Bdale Garbee, Linux CTO, HP
Danese Cooper, Open Source Diva, Intel Corporation

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

LINUX is a registered trademark of Linus Torvalds.

PRINTED WITH SOY INK®

Green Servers

Introducing the iX-Green Neutron Server Line

iXsystems leverages advanced technology and system design expertise to minimize power consumption in all of our servers, making us a leader in power saving technology. Helping companies minimize their carbon footprint is a primary server design objective that results in a significant reduction in the Total Cost of Ownership (TCO) of our systems, through savings on energy and cooling costs.

Given this set of initiatives, iXsystems is proud to introduce the iX-Green Neutron line of environmentally friendly servers within the iX-Earth series. The iX-Green Neutron server line features power-efficient 2.5" hard disk drives, low-power Intel® Xeon® Processor 5500 Series CPUs, Low-Voltage Memory, and the world's first 1U silver level power supply with 86%+ power efficiency. The 2.5" SAS and/or SATA hard drives featured in the iX-Green Neutron line server configurations can reduce power consumption by 40% over standard 3.5" hard drives, and Solid State Drives (SSD) equate to a power savings of up to 95% over standard 3.5" spinning drives.

Improved power efficiency does not mean compromised performance!

The iX-Green Neutron server line is optimized for high performance applications. The **iX-GN1208 1U** configuration features up to 8 cores, up to 96GB of low voltage RAM, and 8 hot-swap 2.5" SAS/SATA drives. The **iX-GN2216 2U** server provides up to 8 cores, up to 144GB of low voltage RAM, and 16 hot swap 2.5" SAS/SATA hard drive bays in an efficient 2U form factor designed for the most frequently encountered applications. The chassis is also equipped with a redundant 720W 1+1 high-efficiency (91%+) redundant power supply for outstanding power savings, and fault tolerance.

The iX-Green Neutron server line is compatible with Intel® Solid State Drives (SSD) for increased performance and even lower power consumption. Unlike traditional hard disk drives, Intel® Solid State Drives have no moving parts, resulting in a quiet, cool storage solution that also offers significantly higher performance than traditional server drives.

The iX-Green Neutron server line within the iX-Earth series is an excellent choice for HPCs, server farms, and other datacenters where space, cost, energy-efficiency, and density are high priorities. By reducing your server(s) power consumption, you will decrease your total cost of ownership (TCO) -- without sacrificing performance -- all while decreasing environmental impact.

iXsystems is the all-around FreeBSD company that builds FreeBSD-certified servers and storage solutions, runs the FreeBSD Mall, and is the corporate sponsor of the PC-BSD Project. For more information about our Green Neutron server line contact iXsystems at (408)943-4100 or visit our website at http://www.ixsystems.com/greenneutron and fill out the inquiry form. One of our expert sales professionals will provide you with a customized quote that best meets your open source hardware solution needs.

800-820-BSDI
http://www.iXsystems.com
Enterprise Servers for Open Source

Intel, the Intel logo, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

iX-GN1208

Features

- Dual 64-Bit Socket 1366 Quad-Core or Dual-Core Intel® Xeon® Processor 5500 Series with QuickPath Interconnect (QPI)
- 1U Form Factor with 8 x 2.5" SAS/SATA Hot-swappable Drive Bays
- Up to 96GB DDR3 1333/1066/800 SDRAM ECC Registered Memory (12 DIMM Slots)
- Left: 1 (x16) PCI-E 2.0 + 1 (x8) PCI-E 2.0 slots + Right: 2 LAN optional (Intel® 82576EB)
- Intel® 82576 Dual Port Gigabit Ethernet Controller
- Matrox G200eW Graphics
- Integrated Remote Management - IPMI 2.0 + IP-KVM with dedicated LAN
- Slim DVD
- 560W high-efficiency power supply (Silver level 86%+)

iX-GN2216

Features

- Dual 64-Bit Socket 1366 Quad-Core or Dual-Core Intel® Xeon® Processor 5500 Series with QuickPath Interconnect (QPI)
- 2U Form Factor with 16 x 2.5" SAS/SATA Hot-swappable Drive Bays
- Up to 144GB DDR3 1333/1066/800 SDRAM ECC Registered Memory (18 DIMM Slots)
- 2 (x16) PCI-E 2.0 slots 4 (x8), PCI-E 2.0 slots (1 in x16 slot), 1 (x4) PCI-E slot
- Intel® 82576 Dual Port Gigabit Ethernet Controller
- Matrox G200eW Graphics
- Integrated Remote Management - IPMI 2.0 + IP-KVM with dedicated LAN
- 5.25" Drive bay
- Slim DVD
- 720W 1+1 Redundant high-efficiency power supply (Gold Level 91%+)

intel Xeon inside™

Powerful. Intelligent.

© sxc.hu/Paula Chhetri

SHAWN POWERS

There's Chocolate in My Peanut Butter

Some combinations are naturally good together: chocolate and peanut butter, toast and jam, macaroni and cheese, pickles and ice cream. Admittedly, the last one was specifically paired by my wife when she was pregnant, but the others seem pretty sound. Sometimes unlikely pairs happen to work really well together too. Open-source applications and the Microsoft Windows operating system is one such pairing.

Before you burn this issue of *Linux Journal* as a symbolic gesture rallying against proprietary operating systems, bear with me for a moment. I'm suggesting we look past the obvious and into the slightly sneaky territory of planting seeds of freedom in otherwise proprietary soil. (Yes, that metaphor made my eyes roll too.) If people start using open-source software in their Windows environments, what is going to keep them from using Windows in the long run? I don't think it will be a love for spyware that keeps them. Cross-platform applications like Firefox, OpenOffice.org and Adobe AIR/Flash have done more to promote the viability of Linux on the desktop than years of me talking about it. This month, we focus on cross-platform development, and for those of us who work in a cross-platform environment, it should be a welcome topic of discussion.

Reuven M. Lerner starts out this issue with the most popular form of cross-platform development: the Web. He shows how to test Rails apps with Shoulda, an interestingly named tool that should help Ruby shine. Marcel Gagné adds to the idea of "open networks" by showing how to utilize open standards on the Internet. He walks us through setting up a Jabber server, which will allow users of any platform to connect and chat. As a warning, if you connect with Marcel, you'll likely end up chatting about Wine—or possibly his funny hat.

Kyle Rankin decided to join the world of Twitter this month, and although there already are many cross-platform Twitter applications, Kyle decided he needed to have it in his little green-on-black text window. Kyle "Mr Twitter" Rankin demonstrates how to make Twitter nothing more than another channel in your IRC client. Deep down, I'm a bit jealous Kyle does most of his communication via IRC, but don't tell him or he'll be impossible to work with.

Although Web applications certainly seem to be the current trend in programming, what if you want a desktop application instead? Mark Obcena shows us Titanium, an open-source platform Web developers can use to create desktop applications. Just like their Web counterparts, Titanium applications allow for cross-platform development.

If Web development isn't your thing, that's perfectly fine too. Mattias Gaertner demonstrates Lazarus for creating platform-independent code. Whether you're aiming for native applications on Linux, Windows or OS X, Lazarus can do it for you. In a similar vein, Johan Thelin tells us about Qt. Although it's most known for its huge role in KDE development, recent versions of Qt integrate quite nicely with GTK+ as well. Add to that cross-platform application support, and Qt continues to be a great development platform.

Don't worry though; here's the paragraph where I tell you it's okay if you don't identify with $ISSUE_FOCUS, because we still have a well-rounded magazine filled with Linux goodies. Mick Bauer dissects Ubuntu's AppArmor and what it means for the security-minded user. Ibrahim Haddad discusses open-source compliance. It would be nice if everyone followed the rules, but sometimes the rules are difficult to understand and the procedures for dealing with them are complicated. Ibrahim helps us out. We also have an interview with the team that is working on Chrome, which is Google's cross-platform Web browser. Yes, I realize it's cross-platform, but with Google's recent announcement of its Chrome OS, Chrome is going to be an entirely new platform of its own! Like all good platforms, however, Chrome, of course, will be based on Linux.

So, if you still think open source has no place in a proprietary world or that cross-platform application development is a bad idea, feel free to burn this issue. While it's burning, you might want to roast a marshmallow over the fire and then combine it with chocolate and graham crackers. That combination *definitely* works.■

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.

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

STARTING AT **$899**

ENTRY LEVEL INTELLIGENT SERVER

SDR-S1341-T00 is among our most cost-effective 1U Xeon Servers, and it is ideal for large high-performance computing deployments

STARTING AT **$959**

APPLICATION SERVER

Refresh your servers with new **SDR-S1337-T02** powered by Intel® Xeon® processor 5500 series, based on intelligent performance, automated energy efficiency and flexible virtualization.

SDR-S1343-T04

STARTING AT **$1,099**

1U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 4X 3.5" HOT-SWAP SATA DRIVE BAYS

- Supermicro 1U Rackmount Server with 560W Power Supply
- Supermicro Server Board w/Intel® 5520 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 4x 3.5" Hot-swap SATA Drive Bays
- Intel® 82576 Dual-Port Gigabit Ethernet Controller

SDR-S2311-T08

STARTING AT **$1,159**

2U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8X 3.5" HOT-SWAP SAS/SATA BAYS

- Supermicro 2U Rackmount Server with 560W Power Supply
- Supermicro Server Board w/Intel® 5500 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 24GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 8x 3.5" Hot-swap SATA Drive Bays
- Dual Intel® 82574L Gigabit Ethernet Controller

SDP-IP308-T10

STARTING AT **$1,599**

PEDESTAL INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 10X HOT-SWAP (OPT.) SATA BAYS

- Intel Pedestal Chassis w/ 750W (1+1) Power Supply
- Supermicro Server Board w/Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg./unbuffered DIMM
- Option 10x 3.5" Hot-swap SATA Bays
- Intel® 8257EB Dual-port Gigabit Ethernet Controller

SDR-S4313-T24

STARTING AT **$1,899**

4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 24X 3.5" HOT-SWAP SAS/SATA BAYS

- Supermicro 4U Rackmount 900W (1+1) Red. Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 144GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 24x 3.5" Hot-swap SATA Drive Bay
- Intel® 82576 Dual-port Gigabit Ethernet Controller

SDR-S3305-T16

STARTING AT **$1,979**

3U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 16X 3.5" HOT-SWAP SAS/SATA BAYS

- 3U Rackmount Server with 1+1 900W Red. Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg.DIMM
- 16x Hot-swap SAS/SATA Drive Bays
- Intel® Dual 82576 Dual-Port Gigabit Ethernet (4 ports)

SDR-C9303-T50

STARTING AT **$4,339**

9U INTEL® XEON® PROCESSORS 5500 NEHALEM SERIES SERVER W/ 50X HOT-SWAP SATA II / SAS BAYS

- 9U Chassis with 1620W Redundant Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 144GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 50 x 3.5"Internal SATA Drives Trays
- Intel® 82576 Dual-port Gigabit Ethernet Controller

10th year

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 to assemble systems that require a little extra. Servers Direct - your direct source for scalable, cost effective solutions.

1.877.727.7886 / www.ServersDirect.com

intel Xeon inside™

Powerful. Intelligent.

Intel, Intel logo, Intel Inside, Intel Inisde logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and Pentium III Xeon are trademarks of Intel Corporation or it's subsidiaries in the United States and other countries.

letters

A Funny

I just found this so funny I had to share. MSN decided to call its search engine Bing. I don't know who the idiot was who thought of this one. I know in Chinese *Bing* means "disease", so Windows is admitting what it is? LOL—I just thought this should be shared with the Linux world.

--

Dean Hill

Linux on the Desktop, Part II

I am sitting here reading the new *Linux Journal* that came today (July 2009), and I come across this letter to the editor from Kulmacet titled "Linux on the Desktop?" where he mentions that Linux as a desktop OS has the apps, but lacks the stability. I, like Shawn in the response, am scratching my head. Typically, users feel the opposite. We have the stability, but lack some of the mainstream apps. He goes on basically to mention how Windoze is quirky but at least it works.

Yes, when you buy a fresh new Dell or HP, it just works. Those companies have taken the time in their shops to make sure it does before shipping. But, how many of us out there have built a fresh new rig with Windows? Does it "just work" then? No. You still have to find/install the newest drivers and hope they have them for your version (Vista, XP, 32-/64-bit). Linux is the same way. Sure, you may have to do a little configuring, but you do that in Windows as well. I am currently using a Qosmio X305-Q705 running Fedora 10 64-bit. And when I did a fresh install of Fedora, guess what? Everything worked! Sound, wireless, video—everything. So "Windows just works" is not a great argument. Most computers running Linux have decent support for most devices out of the box. The most I really have to do is occasionally get some wireless drivers or the NVIDIA drivers (which have a Linux version) if I intend to do any heavy gaming. For business, you probably wouldn't need this.

And, regarding those forums mentioned in the letter, in *most* cases, those unanswered problems probably were answered in many other places in those same forums. It just takes a little looking. For "typical" users, how many Blue Screens of Death do you see in Windows compared to kernel panics with Linux? Linux surpasses Windows in stability by far, no matter your experience level—just my two cents though.

--

Cary

My experience has been similar to yours. In fact, at first I thought the letter was tongue in cheek, but it looks like the reader had some serious bad luck with Linux. Hopefully, our responses will encourage folks having difficulties with stability to take another look at their hardware, because as you mention, stability is generally one thing Linux gets very right.—Ed.

Unsung Hero

Has anyone thought to check out the World Digital Library? Per the publication *The Library of Congress Gazette* article published today (May 29, 2009) titled "WDL 1.0 Technical Info", the following might be of interest to the Linux community:

- Development time: ~13 months
- Lines of code: ~50,000
- Test cases written: ~600
- Development platform: Linux
- Deployment platform: Solaris
- Key technologies: Django, Python, MySQL, Solr/Lucene, Squid Nginx, Seadragon

It also provided the following launch-day statistics:

- Page views: 7.1 million
- Visitors: 600,000
- Peak hits/hour: 32 million

When there is a major application being contemplated and the IT folks all say Windows, maybe this will give them pause. The article did not give a byline or mention which distribution of Linux, unfortunately.

--

Paul F. Baltrunas

Sadly, this is one more example of the unsung heroics Linux is responsible for accomplishing. I wish "open-source infrastructure" was a required course for anyone going into the IT field. Unfortunately, the implementation of open source generally is driven by (lack of) finances. Thanks for the information.—Ed.

Disappointed with KDE 4

I recently upgraded from Kubuntu 7.10 to Kubuntu 8.10, and I was amazed to find that KDE had jumped from version 3.5 to version 4.2! I am a really KDE-ish guy, as the first Linux desktop I ever saw was KDE 3.x on Knoppix. As a result, I learned to love KDE. KDE feels like home. I don't hate GNOME. GNOME is a great desktop, and it even starts and runs faster than KDE. But, it is less configurable than KDE, and even though the difference is minor, it's enough for me to use KDE—at least, until KDE 4.x came along.

KDE 4.x is a major recode from the 3 series. For instance, the desktop now has a widget-based setup—widgets go on the desktop, not files. You want a comic strip on the desktop? Sure, it has a widget for that. You

want a battery monitor? A dictionary? A clock? A calculator? A 15-piece puzzle? It's got widgets for those too. How exciting—until you remember that "widgets, not files" has "NOT FILES" in it. No more putting icons directly on the desktop, which is great, if you always have wanted a comic strip on the desktop. But, no files directly on the desktop isn't the only bad surprise that comes packed with KDE 4. It also is much less configurable, because *so* much was recoded, the KDE coders have not had time to re-add all the functions of good-old KDE 3, and the configuration options that go with them are all missing—for example, auto-hiding the Kicker. You can't auto-hide the Kicker in KDE 4 at all, whereas KDE 3 could be set up to hide the Kicker as soon as the mouse left the Kicker and show it again as soon as the mouse hit the bottom of the screen.

Of course, there is a reason for all this. Apparently, the KDE coders felt KDE was getting behind the times, and it was time to upgrade the interface. KDE 4 has good desktop effects, and although I love the effects and interface, for now, KDE does it at the expense of everything else.

Are you ready to give up your favorite file manager, Konqueror, for a new file manager that is missing some of Konqueror's best features? Then, upgrade to KDE 4.

Hopefully, the KDE guys didn't give up good features for good. I'm waiting on edge for KDE 5, in which, hopefully, all of KDE 3's good features will have returned. But for now, I am using GNOME, which still has much more reasonable features.

--
Christian H.

From Afghanistan with Love

I'm currently in the hills of Afghanistan and found that receiving snail mail is very unpredictable/unreliable. Therefore, being new to the Linux world, I was looking for quality reading that I could download onto my laptop. After searching all the Web, I came to the conclusion that your subscription helped me with the understanding of how Linux operates compared to the dreaded Windows and Mac environments. I've completely removed all Windows from my system now. Special thanks go out to the support in your subscription department as well! I accidentally subscribed to the print edition, and within minutes called and received a subscription change. This allowed me to download the latest digital edition of your magazine. Additionally, I was able to go to the back editions of *Linux Journal* and grab all the ones that interested me. Again, you guys ROCK! Thanks for the great service and product!

--
Stephen Alderete

That's great to hear! Thanks for sending us a note, and if you have a reliable connection to the Internet, be sure to visit our Web site as well. There are lots of things on-line that don't make it into print.—Ed.

Ever Mangle a Configuration File?

Reading the December 2008 issue (five months late!), I was somewhat amused by the box on page 37 "Regenerating smb.conf in Debian/Ubuntu". Re-installing a package merely because a configuration file got mangled seemed rather unnecessary. If I am experimenting with something new, I will keep a copy (such as smb.conf-orig). Beyond that, I keep configuration files under RCS control, so I can turn back the clock to any version I want.

--
David Penman

You certainly have best practice in mind when you tweak your config files, but unfortunately many users do not. Sadly, I often fall into that category myself! And don't get me started on how many times I have to ask users, "Do you have a backup?" Thankfully, Linux distributions generally have a way to get back to the defaults when we do silly things.—Ed.

Linux on the Desktop, Part III

In the July 2009 issue's Letters section, Kulmacet commented that Linux was still not a good desktop OS and did not "work" out of the box. When I hear this, I just scratch my head. Maybe the reader was installing an older Gentoo? Ubuntu, OpenSUSE and Fedora are all mainstream distros and install very easily. Recently, a family member asked me to take a look at her computer that took ages to boot up, and then was so slow it was pretty much unusable. I don't have to go into the sordid details of all the viruses, spyware and other junk the scanner turned up. "Can someone just make a system that works and doesn't get all these viruses?", she asked. I backed up the data, wiped the drive and installed OpenSUSE 11.1 with GNOME. After a brief tutorial, off she went. A few months later, I hadn't heard anything and had assumed Windows was re-installed. No, she was very happy, and the system was fast with no viruses. She even installed a new printer, scanner and camera.

Not every user will have this experience, but I have converted quite a few friends and coworkers to Linux during the past few years. Of course, I use my knowledge and experience to get them over that fear of the unknown. I am sure that your readers could share similar experiences.

--
George

Like my previous comment, I fully agree with you. The reader last month obviously had an uncommon, and unfortunate Linux experience. Hopefully, we'll all be the encouragement needed to try again!—Ed.

No More Break-Ins

Nice article on the WD MyBook World Edition [see Federico Lucifredi's "Hacking Your Portable Linux Server" in the July 2009 issue]. I just picked up a copy from the local computer store and was happy to discover that one doesn't have to break in anymore. The WD software allows you to open up the system nowadays. I don't know if that works for all World Editions—in particular, the MyBook II in the article—however, my guess would be it works there too.

--
Hans Kramer

Re: Bad Guys

I let the first letter to the magazine that disturbed me go, but after reading this most recent complaint, I had to write in. However, I in no way am attacking *Linux Journal*. As a communist and member of the International Socialist Organization, I always find it saddening to read or hear people's distorted understanding of communism. Francis Kohl wrote in to claim Marx is responsible for the "most horrible dictatorships in history" [see the July 2009 issue's Letters]. There are two reasons this is usually a view pushed by people. First, because Marx called for the "dictatorship of the proletariat"; however, this in no way refers to an individual dictator. It is the ruling of the entire proletariat over the bourgeoisie. Second, because many individual dictators have proclaimed themselves Marxists and even have socialist states. But, someone saying something

does not make it true, as we see with Kohl's comments.

If you care to blame someone for Pol Pot's abuses, why not blame someone who *did* influence him? Like Buddha? Pol Pot's Theravada Buddhism is what created his idea of "communism". This Buddhism is why he saw the rural peasants as the revolutionary class and the need to push everyone out of the cities and massacre the "unnecessary" part of the population. This is directly opposed to Marxism!

Stalin was a horrible dictator, and Lenin saw this potential and while sick and dying, called for his removal from power. Stalin already had gained much support in the military by this time, and years later began the terror against those in the party opposing his disgusting regime. Thus, he executed and exiled the few remaining communists, like Leon Trotsky, who escaped to Mexico before being assassinated by a Stalin hit man.

I'm also confused at the fact that no one on the left or right who condemns Marx for these atrocities ever blames capitalist thinkers for the dictators who had "free markets" (like ours), mass privatization and cuts on welfare programs—Augusto Pinochet of Chile, for example. They also don't attack capitalist thinkers for the deaths of millions due to the quest for profit followed by the corporations around the world. This is not to say Marx and Lenin were pacifists or perfect. Both were human and made mistakes in their ideas and actions, and both understood the ruling class would not go away without a fight. If Kohl opposes this, I assume he does not dare to stand to the United States' National Anthem—you know, that song about a bloody revolutionary to overthrow the ruling class and enact progressive measures?

Lastly, there was a letter many months ago [January 2009], in which Gene said he wanted students to learn more Adam Smith and less Karl Marx. I think Gene may want to study some Adam Smith first. Adam Smith was a supporter of a progressive income and estate tax on the rich to "contribute to the public expense"! Read *The Wealth of Nations*, and you'll see it's clear Smith would be condemned as a socialist if he were a politician in the United States today.

I am always shocked at the large anti-socialist crowd in the Free Software community, when I see the Internet age and Free Software as great examples of the potential for communism. Communism is not a bullet in the back of your head for not sharing, as Eric S. Raymond said in *Revolution OS*. I'd be more worried about him and his collection of guns shooting someone for being on his private property.

--
Tristan Sloughter

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.

Even though Tux didn't enroll in the US Army, he still has to serve a one-year deployment in Iraq with me. It's a little bit too hot for him also! Submitted by SPC Dumitru Sly Silviu-Cristian, HHC 1-63 CAB 2HBCT 1ID Scout Plt, Camp Stryker.

LINUX JOURNAL

At Your Service

MAGAZINE

PRINT SUBSCRIPTIONS: Renewing your subscription, changing your address, paying your invoice, viewing your account details or other subscription inquiries can instantly be done on-line, **www.linuxjournal.com/subs**. Alternatively, within the U.S. and Canada, you may call us toll-free 1-888-66-LINUX (54689), or internationally +1-818-487-2089. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 16476, North Hollywood, CA 91615-9911 USA. Please remember to include your complete name and address when contacting us.

DIGITAL SUBSCRIPTIONS: Digital subscriptions of *Linux Journal* are now available and delivered as PDFs anywhere in the world for one low cost. Visit **www.linuxjournal.com/digital** for more information or use the contact information above for any digital magazine customer service inquiries.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at **www.linuxjournal.com/contact** or mail them to Linux Journal, PO Box 980985, Houston, TX 77098 USA. Letters may be edited for space and clarity.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line, **www.linuxjournal.com/author**.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line, **www.linuxjournal.com/advertising**. Contact us directly for further information, ads@linuxjournal.com or +1 713-344-1956 ext. 2.

ON-LINE

WEB SITE: Read exclusive on-line-only content on *Linux Journal*'s Web site, **www.linuxjournal.com**. Also, select articles from the print magazine are available on-line. Magazine subscribers, digital or print, receive full access to issue archives; please contact Customer Service for further information, subs@linuxjournal.com.

FREE e-NEWSLETTERS: Each week, *Linux Journal* editors will tell you what's hot in the world of Linux. Receive late-breaking news, technical tips and tricks, and links to in-depth stories featured on **www.linuxjournal.com**. Subscribe for free today, **www.linuxjournal.com/enewsletters**.

EtherDrive®

The AFFORDABLE Network Storage

Now Supporting VMware® ESX 3.5

Fibre Channel speeds at Ethernet prices!

Is your budget shrinking while your network storage needs are growing? Are you suffering from "sticker shock" induced by expensive Fibre Channel and iSCSI storage area network solutions? EtherDrive® SAN solutions offer Fibre Channel speeds at Ethernet prices! Starting at just $1,995 for a 4TB system, EtherDrive® is the ***affordable*** storage area network solution. With sustained access speeds from 200MBytes/sec to over 600MBytes/sec, EtherDrive® SAN solutions are ***fast***. From a 4TB single storage appliance to multi-PetaByte system by simply adding more storage appliances, EtherDrive® SAN solutions are ***scalable***. From a single storage appliance to a network of sophisticated virtualized storage LUNs, EtherDrive® SAN solutions embrace ***virtualization***.

Coupling Ethernet technology with SATA hard disk drives, EtherDrive® SAN solutions exploit commodity components to deliver ***affordable, fast*** storage area network solutions that keep more green in your wallet! Whether you use your own SATA compliant disk drives or our certified enterprise class disk drives, you are in control! EtherDrive® SAN solutions accept standard SATA hard disk drives. Ethernet and SATA disk drives - two proven technologies in one ***affordable, fast*** storage area network solution - EtherDrive®.

EtherDrive® SAN solutions use the open ATA-over-Ethernet (AoE) lightweight network storage protocol. Simple. Easy to understand. Easy to use. AoE uses Ethernet to transport ATA disk commands without the burden of TCP/IP overhead, thereby enabling disk drives to become AoE devices connected directly to an Ethernet network. An AoE device can be a single physical disk or a logical device made up of multiple disks. An EtherDrive® SAN appliance is an AoE target device.

Finally, an ***affordable, fast*** storage area network solution for your VMware® ESX 3.5 installation. The EtherDrive® VMware ESX Host Bus Adapter empowers ESX with AoE technology to deliver EtherDrive® SAN solutions for your VMware ESX 3.5 installation.

Shipping EtherDrive® RAID solutions since 2004, Coraid boasts thousands of satisfied customers spanning a broad spectrum of the market including enterprise, government, educational institutions, and hosting service providers. ***Call today*** to order your EtherDrive® solution, and join the ranks of our thousands of satisfied customers!

Call 1.877.548.7200
or visit our website at
www.coraid.com
International: +1.706.548.7200

CORAID

vmware® | technology alliance PARTNER

ESX 3.5 compatible EtherDrive® HBA

© 2009 Coraid Inc. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.
All other marks and names mentioned herein may be trademarks of their respective companies.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Rik van Riel has doubled and doubled again the amount of **RAM** that can be directly addressed in the x86 64-bit architecture. The previous limit had been 2^{44} bytes, or more than 17 terabytes. The new limit is 2^{46} bytes, or more than 70 terabytes.

The **Linux Pulse Per Second** (LinuxPPS) Project has had to reset and restart, when **Udo van den Heuvel** asked why the code hadn't been accepted, and neither **Andrew Morton** nor **Alan Cox** could remember any of the objections anyone had against it. They both recommended resubmitting the patches, which at the very least would get the folks who still had problems with the code to speak up again. LinuxPPS is a project to provide a character-device-based API for communication between kernel space and userspace. **Rudolfo Giometti** took Alan and Andrew's advice a couple weeks later, submitting the core LinuxPPS code for inclusion—the idea being to get everyone signed off on the basic features before introducing any code that might be more controversial. He also pointed out that all previous objections had been fixed, or that the objectors already had agreed the fix could wait. So, it looks like a good thing that Udo asked about this initially, or the perfectly good code might be lingering still.

The **XCEP motherboards** from **IskraTel** now are supported in Linux, which is cool, because that motherboard is used in many particle accelerators throughout the world. **Michael Abbot** recently submitted patches adding this architecture, which runs an ARM XScale PXA255 CPU.

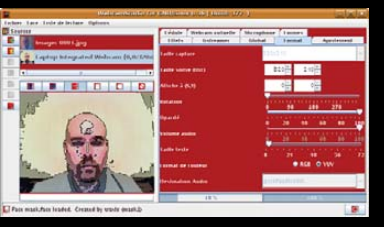
DebugFS soon may be configurable in much more powerful ways. **Steven Rostedt** has added a feature to enable tracing events defined in a whole directory tree. The previous version required that each event be enabled individually in its own directory. The current version recurses through all child directories, but it also allows users to chop off branches of that directory tree easily if they so desire. What's the cost of all this power? It's no longer easy to identify which tracing events are enabled and which are not, because an event may be controlled by configurations elsewhere in the directory tree. But, as Steven said during the discussion, the information is all there, and a script easily could identify all configured events. As far as the debate went, no one seemed to feel the cons outweighed the pros, so this probably will be accepted into the kernel in the near future.

One thing that doesn't happen often is a hardware vendor asking for advice from the Linux community about how to code its drivers. But, **Atul Mukker** from **LSI Corporation** recently did exactly that. He said LSI wanted to take a whole new approach to driver writing, in which it had operating-system-independent code at the core, with a thin layer of support for Linux, Windows and so on. And, he just wanted to know if anyone had any advice. Turns out several folks did—one of the main ones being **Jeff Garzik**. Jeff recommended **Intel's networking drivers** as excellent examples of good practice. He suggested modularizing the code so that each piece of hardware would have its own codebase, which also could be kept free of any operating-system-specific code. He also recommended keeping general-purpose code out of the driver entirely, where other drivers could use it more easily. The **Application Binary Interface** (ABI), Jeff said, also should remain consistent with other drivers already in the kernel. Any feature similar to something found elsewhere should imitate that other interface. Any features that were unique, on the other hand, could create whatever interface seemed best.

—ZACK BROWN

WebcamStudio—Create Your Own On-line Video Show

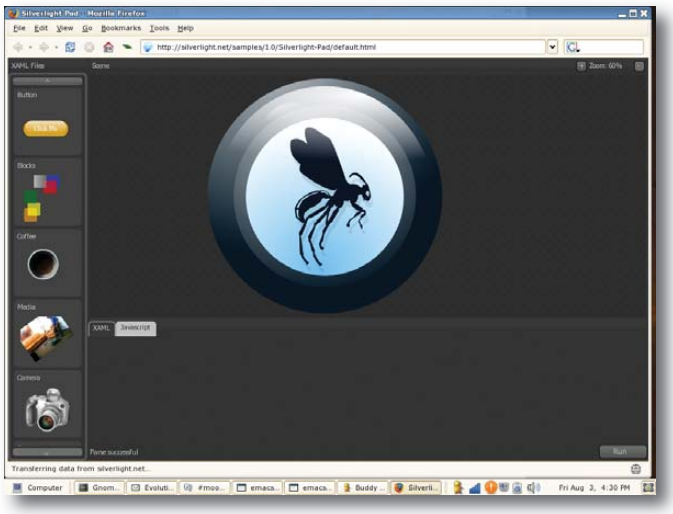
A few months back, *Linux Journal* had a live streaming show called, "*Linux Journal* Live". It aired once a week and streamed via ustream.tv. One of the frustrating things about running the show was that it was very difficult to get the "studio" feel using Linux. As it happened, we ended up using a Macintosh computer and the freeware CamTwist in order to embed graphics, guest hosts and text.



If we ever resurrect the live show, now we'll be able to stream from our dearly beloved Linux, thanks to the open-source project, WebcamStudio (**webcamstudio.sourceforge.net**). WebcamStudio allows Linux users to stream Webcams, graphics, text and much more to sites like ustream.tv. If you've ever wanted to try your hand at a live show, be sure to check it out.

—SHAWN POWERS

NON-LINUX FOSS



Silverlight Pad Running on Moonlight (from www.mono-project.com)

Moonlight is an open-source implementation of Microsoft's Silverlight. In case you're not familiar with Silverlight, it's a Web browser plugin that runs rich Internet applications. It provides features such as animation, audio/video playback and vector graphics.

Moonlight programming is done with any of the languages compatible with the Mono runtime environment. Among many others, these languages include C#, VB.NET and Python. Mono, of course, is a multiplatform implementation of ECMA's Common Language Infrastructure (CLI), aka the .NET environment.

A technical collaboration deal between Novell and Microsoft has provided Moonlight with access to Silverlight test suites and gives Moonlight users access to licensed media codecs for video and audio. Moonlight currently supplies stable support for Silverlight 1.0 and Alpha support for Silverlight 2.0.

—MITCH FRAZIER

LinuxJournal.com

As we read this month's coverage of cross-platform development, I thought I'd weigh in on the Web development end of things. While I work toward a new-and-improved iteration of LinuxJournal.com, I must constantly consider the needs of users with widely varying operating system and browser configurations. LinuxJournal.com visitors are a technologically diverse bunch. As you might expect, the majority of our Web visitors view LinuxJournal.com with Firefox, but what may surprise you is that a slight majority of those Firefox users are browsing from a Windows machine. Linux and Firefox users are nipping at their heels though. What also may surprise you is the percentage of visitors browsing with some version of Internet Explorer. Granted, that percentage has decreased during the last couple years, but the most recent numbers show about 20% of traffic coming from IE users, down from around 30% a year ago. Other browsers like Chrome, Opera and Safari have a small but important constituent as well, which makes my job just a little more interesting. So, to all of you visiting us from a less-used browser, I am doing my very best to give you the same great experience as the Firefox majority, and to all of those using IE, well, you may drive me to drink. I still welcome you though, and I will do my best to accommodate!

—KATHERINE DRUCKMAN

LJ Index September 2009

1. Percent of all waste that is e-waste: **2**
2. Percent of the heavy metals in landfills that come from e-waste: **70**
3. Number of separate elements found in e-waste: **38**
4. Percent of e-waste bound for recycling that actually gets recycled: **20**
5. Average number of electronic items purchased per American household per year: **24**
6. Average number of books read per year by adults in the US: **4**
7. Percent of adults in the US that read zero books per year: **25**
8. Number of hours the average American spends watching TV per day: **4**
9. Number of years spent watching TV during a 65-year life: **9**
10. Average time someone in the US spends Web surfing each month: **27:38:58**
11. Average time someone in France spends Web surfing each month: **19:16:28**
12. Average time someone in Spain spends Web surfing each month: **17:52:43**
13. Average time someone in the UK spends Web surfing each month: **17:36:55**
14. Average time someone in Germany spends Web surfing each month: **17:00:35**
15. Average time someone in Italy spends Web surfing each month: **15:02:36**
16. Average time someone in Australia spends Web surfing each month: **14:30:16**
17. Percent of local advertisers on search engines that choose not to renew: **50**
18. Percent of local advertisers on advertising sites that choose not to renew: **60**
19. US National Debt as of 06/08/09, 10:51:06am MST: **$11,403,815,042,547.90**
20. Change in the debt since last month's column: **$152,944,501,331.18**

Sources: *1–3: EPA / 4: Basel Convention / 5: Consumer Electronics Association / 6, 7: Washington Post / 8: A.C. Nielsen Co. / 9, 20: Math / 10–16: Telegraph.co.uk / 17, 18: The Business Insider / 19: www.brillig.com/debt_clock*

Mac OS X, It's Not Linux, but It's Close

In the past, the Mac OS was a fairly unique entity, not having much in common with other OSes, such as Windows or UNIX, which made cross-platform work a bit convoluted. However, the advent of the latest incarnation of the Mac OS, called OS X or Darwin, provides a very comfortable alternative for Linux geeks. Because Darwin is based on BSD UNIX, it is possible to use POSIX-compliant applications on the Mac.

Apple provides a package called Xcode on its developer site. Xcode has the necessary tools for compiling programs on the Mac, and it includes a nice graphical IDE and lots of examples for developing applications for OS X. Xcode is based on the GNU toolset, providing tools like gcc, libtool, make and so on. That means, with Xcode, most command-line applications can be compiled and run on the Mac. So, a simple little hello world program:

```
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char **argv) {
  printf("Hello World\n");
}
```

compiles fine with gcc, giving you an executable that prints out "Hello World" on the command line. Basically, anything that is POSIX-compliant should compile and run with no issues.

Getting graphical programs to run can be a bit more involved. Mac OS X does provide an X server and all the standard development libraries you would need for a pure X11 application, like Xlib. However, none of the other standard libraries, like GTK or Qt, are available by default. You have to download, compile and install them yourself, which works fairly well, but you have to choose the correct configuration options and collect all the required dependencies. But, you shouldn't need to go through so much pain. Two projects in active development provide some form of package management for GNU software: Fink and MacPorts. Using these, getting and installing GNU software is as easy to do as it is with most Linux distros.

The Fink Project started in 2001 and is based on the Debian package management system, so you can use the Debian package tools like dpkg, dselect and apt-get, making it familiar for Debian-based distro users. Once the base installation is done, you can start to install packages. If you like a text-based manager, use dselect (Figure 1). If you prefer a graphical manager instead, use the following command to get synaptic (Figure 2):

```
sudo apt-get install synaptic
```

Using these applications, you can install many of the packages you are familiar with in Linux. The package count, at the time of this writing, is 10,872.

However, not all packages are available as a binary install using these tools. For that class of packages, Fink installs them directly from source, compiling and installing on your Mac. So, for example, if you want to install gramps and do some genealogy work, execute the following:

```
sudo fink install gramps
```

Even installing from source, Fink deals well with dependency issues, because it still is based on the Debian package management system.

The MacPorts Project started in 2002 and models itself after the BSD port packaging system. Thus, you use the command to manage the packages on your system. Once you have done the base install, you can install other software packages simply by running the command:

```
sudo port install stellarium
```

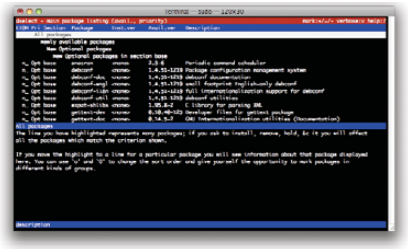


Figure 1. dselect Package Manager

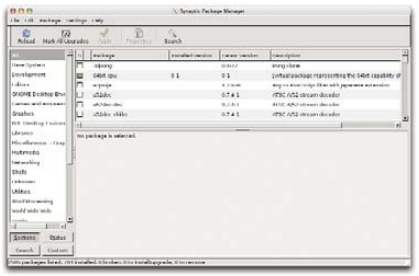


Figure 2. synaptic Package Manager

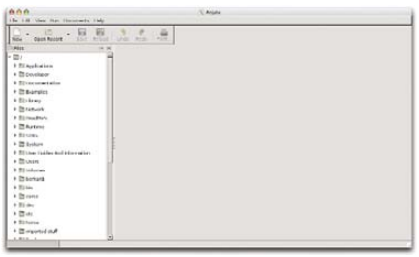


Figure 3. Anjuta IDE

Several graphical interfaces are available as well, such as Porticus. However, those typically are independent projects, as opposed to the Debian tools available in Fink. As such, their development cycle and behavior tend to be a bit more erratic and unstable than the older and more mature Debian tools. But still, they may be exactly what you're looking for if you prefer a graphical interface. Like the Fink Project, both binary packages and source packages are available. There are 5,829 packages available in the MacPorts Project.

Both projects provide access to the full wealth of open-source applications that has been available to Linux users, and the number of packages provided by both projects continues to grow.

Once you have one, or both, of these projects installed (they will coexist on your system), you will have all the tools necessary to do your own code development. I have used anjuta (Figure 3) on my MacBook to develop some small GNOME applications. These compile and run equally well on my MacBook and my Netbook running Ubuntu. Although there isn't binary compatibility between OS X and Linux, with source compatibility, it is (at least in theory) simply a matter of recompiling for the other system.

Running Mac OS X code on Linux is not as easy as running Linux code on Mac OS X. The real stumbling block is the graphical interface called Quartz on the Mac OS. Although the kernel and most of the command-line tools have been released as open-source software, Quartz still is closed. At the time of this writing, I could not find any references to a reverse-engineered, open-source replacement for Quartz. So the only option available is running OS X inside a virtual machine. Although this is not technically running Mac applications on Linux, it does provide the ability to run OS X on a Linux box.

—JOEY BERNARD

Resources

Apple Developer Connection: **developer.apple.com**

Open-Source Apple: **www.opensource.apple.com**

Fink Project: **www.finkproject.org**

MacPorts Project: **www.macports.org**

Why Buy a $350 Thin Client?

On August 10, 2009, I'll be at a conference in Troy, Michigan, put on by the LTSP (Linux Terminal Server Project, **www.ltsp.org**) crew and their commercial company (**www.diskslessworkstations.com**). The mini-conference is geared toward people considering thin-client computing for their network. My talk will be targeting education, as that's where I have the most experience.

One of the issues network administrators need to sort out is whether a decent thin client, which costs around $350, is worth the money when full-blown desktops can be purchased for a similar investment. As with most good questions, there's really not only one answer. Thankfully, LTSP is very flexible with the clients it supports, so whatever avenue is chosen, it usually works well. Some of the advantages of actual thin-client devices are:

1. Setup time is almost zero. The thin clients are designed to be unboxed and turned on.
2. Because modern thin clients have no moving parts, they very seldom break down and tend to use much less electricity compared to desktop machines.
3. Top-of-the-line thin clients have sufficient specs to support locally running applications, which takes load off the server without sacrificing ease of installation.
4. They look great.

There are some advantages to using full desktop machines as thin clients too, and it's possible they will be the better solution for a given install:

1. Older desktops often can be revitalized as thin clients. Although a 500MHz computer is too slow to be a decent workstation, it can make a very viable thin client.
2. Netbooks like the Eee PC can be used as thin clients and then used as notebook computers on the go. It makes for a slightly inconvenient desktop setup, but if mobility is important, it might be ideal for some situations.
3. It's easy to get older computers for free. Even with the disadvantages that come with using old hardware, it's hard to beat free.

Thankfully, with the flexibility of LTSP, any combination of thin clients can coexist in the same network. If you're looking for a great way to manage lots of client computers, the Linux Terminal Server Project might be exactly what you need. I know I couldn't do my job without it.

—SHAWN POWERS

They Said It

We're done with the first 80%, and well into the second 80%.
—Larry Wall, referring to Perl 6

Doing linear scans over an associative array is like trying to club someone to death with a loaded Uzi.
—Larry Wall

Getting information off the Internet is like taking a drink from a fire hydrant.
—Mitchell Kapor

Globalization, as defined by rich people like us, is a very nice thing...you are talking about the Internet, you are talking about cell phones, you are talking about computers. This doesn't affect two-thirds of the people of the world.
—Jimmy Carter

I don't have to write about the future. For most people, the present is enough like the future to be pretty scary.
—William Gibson

In Cyberspace, the First Amendment is a local ordinance.
—John Perry Barlow

Hardware Requirements: None

In two days, I'll be the proud owner of a Kindle DX. That may seem a bit odd, considering how much I despise DRM. The real selling point for me, however, is that it will read PDF files natively, and in full size. As I was looking for the system requirements for the Kindle DX (naïvely thinking it might sport Linux support), I was amused to see the hardware requirements listed: none.

The Kindle is designed as a self-contained piece of hardware, never needing to connect to a computer. Because it actually mounts as a USB removable device, it will work just fine under Linux. But, more interesting for me is that it never needs to sync at all. And, that got me thinking about my other electronic devices. I have two smartphones that I never connect to a computer. They both have the ability to sync with a computer, but because they're connected to the Internet, I never have had the need to connect them directly to a computer.

Will hardware compatibility fade away into the past? It wouldn't be a bad thing, unless, of course, proprietary hardware drivers are replaced with proprietary network protocols. Luckily, Linux is king on the Internet, so we're much more likely to keep standards in place on-line than in the hands of Windows-savvy developers.

My Kindle DX might have the taint of DRM, but thankfully, it also has support for non-DRM files as well. Although it has support for the non-free Windows operating system, it also supports Linux. And heck, it will run just fine all by itself. I figure that's because it's running Linux as its underlying OS.

—SHAWN POWERS

Testing Rails Applications with Shoulda

REUVEN M. LERNER

New to testing? Just want an easier time with Test::Unit? Shoulda is the answer.

The past few months, I've been looking at a number of tools that make it easier for Ruby on Rails developers to improve the reliability of their software using automated testing. Even if you don't fully subscribe to the notion of test-driven development (TDD) or its cousin, behavior-driven development (BDD), the fact that Rails makes it so easy to test each part of your code makes it less likely that foolish mistakes will creep into your applications.

By default, Rails comes with Test::Unit, a test suite that makes it possible, and even easy, to check your code. Coupled with the test classes that come with actionpage, one of the core Ruby gems that comes with Rails, you can create a comprehensive test suite at the unit (model), functional (controller) and integration (cross-controller) levels. If you have a comprehensive test suite, you easily will detect, and understand the implications of, changes you make to the code that break the test.

That said, Test::Unit sometimes can be a bit verbose and repetitive. If you are writing unit tests, and you want to make sure that a particular attribute has been tested completely, it would be nice to be able to express a number of test cases quickly and tersely. Tests can function, in many ways, as a type of specification (as I will explain when we get to RSpec and Cucumber in coming months), and the easier it is to read these specifications, the less likely odd behavior is to slip through the cracks. It also goes without saying that the easier it is to write tests, and particularly comprehensive tests, the more likely you are to write them.

This is why Shoulda, a set of macros that work with Test::Unit, has become popular among Ruby developers in general and Rails developers in particular. Shoulda, developed by Tammer Saleh, a programmer who works for the Thoughtbot consulting company in Boston, is a set of macros that makes it easier to write tests with Test::Unit, as well as easier to read them. I have begun to use Shoulda with projects that I test with Test::Unit and have found it to be quite enjoyable.

This month, I take a look at Shoulda, and how you can integrate its macros into the tests you write in a Rails application. I explain how Shoulda divides tests into contexts, allowing you to group tests together even within a single file. I also describe how Shoulda's various macros make it easy to run a number of tests using a single readable line.

I should note that although Shoulda originally was designed to be used with Test::Unit and to provide something of an RSpec-like environment for Test::Unit users, it adds a growing amount of support for RSpec as well. Even if you use RSpec, you might want to consider using Shoulda along with your standard RSpec tests (or specs). I haven't looked at the combination for my own work, but it might be appropriate for what you're doing.

Installation and Basic Use

Shoulda comes packaged as a Ruby gem, and can be installed as:

```
sudo gem install thoughtbot-shoulda --source=http://gems.github.com
```

Earlier versions of Shoulda came packaged under a slightly different name (Shoulda, rather than thoughtbot-shoulda). It also is possible to install Shoulda as a Rails plugin; in this article, I assume that you have installed the gem version.

You can incorporate the gem in your configuration file, config/environment.rb:

```
config.gem "thoughtbot-shoulda", :lib => "shoulda",
➥:source => "http://gems.github.com"
```

With that in place, your Rails application either will run with Shoulda in place, or it will fail to load, complaining that the gem has not been installed. In one of my favorite Rails functions, you then can type:

```
rake gems:install
```

and your Rails application will examine its list of required gems, download those that are not yet on the system and install them in the appropriate places.

Let's assume you have created a simple Rails application that contains a single model that describes people. You can create it in the

following way:

```
rails simple
cd simple
./script/generate model Person firstname:text lastname:text
 ➥birthdate:date grade_in_school:integer phone_number:text
 ➥email_address:text
rake db:migrate
```

At this point, you now have a simple Rails application (using the built-in default database, SQLite) with a single model defined. By creating your model with a generator, you get the following simple unit test file:

```
require 'test_helper'

class PersonTest < ActiveSupport::TestCase
  # Replace this with your real tests.
  test "the truth" do
    assert true
  end
end
```

True, you can invoke `rake test` on this, and the tests will succeed, but that's because the test is completely empty. You can write:

```
rake test:units
```

but the success won't really tell you much, other than the fact that you need to write some tests.

Testing with and without Shoulda

Now comes the hard part. What sorts of tests do you want to write? Well, that depends on the constraints you have put on your model, typically by using ActiveRecord validations.

Specifically, you presumably will want to make sure that the people have a first and last name, and that their grade in school (for the purposes of demonstrating some additional testing) is greater than 0 and less than 13. You will want to make sure that the person's birth date is in the past. You also will want to make sure that every e-mail address in the system is unique to avoid having more than one person with the same e-mail address.

In the model file itself, the validations will look like this:

```
class Person < ActiveRecord::Base
  validates_presence_of :firstname, :lastname, :email_address
  validates_uniqueness_of :email_address
  validates_numericality_of :grade_in_school,
  ➥:greater_than_or_equal_to => 0, :less_than_or_equal_to => 13
end
```

If you simply were using Test::Unit, you probably would want to test each of these validations. This has less to do with testing the validations and more to do with ensuring that your code meets the specifications you have laid out. (If tests were only a means of checking the correctness of your code, you could make a pretty good argument against tests for these validations, because ActiveRecord already has a fairly extensive test suite.)

If you were to try to test this line:

```
validates_presence_of :firstname, :lastname, :email_address
```

you would need to iterate over each of the three fields that are mentioned, checking to see whether

Listing 1. person_test.rb

```
require 'test_helper'

class PersonTest < ActiveSupport::TestCase
 # Replace this with your real tests.
 test "working person" do
   person = Person.new(:firstname => 'First',
                       :lastname => 'Last',
                       :email_address => 'foo@example.com',
                       :grade_in_school => 10)
   assert person.valid?
 end

 test "person must have first name" do
   person = Person.new(:firstname => '',
                       :lastname => 'Last',
                       :email_address => 'foo@example.com',
                       :grade_in_school => 10)
   assert !person.valid?
 end

 test "person must have last name" do
   person = Person.new(:firstname => 'First',
                       :lastname => '',
                       :email_address => 'foo@example.com',
                       :grade_in_school => 10)
   assert !person.valid?
 end

 test "person must have e-mail address" do
   person = Person.new(:firstname => 'First',
                       :lastname => 'Last',
                       :email_address => '',
                       :grade_in_school => 10)
   assert !person.valid?
 end
end
```

the model would be valid if one of these were missing. See Listing 1 for an example of what person_test.rb, the file that contains the unit tests for the Person object, would look like just to test the need for each of those.

But, you lose something in creating these verbose tests. Instead of functioning as a checkup on your code, and as a specification of sorts for what you intend to do, these tests become verbose, repetitive and difficult to read.

With Shoulda installed, you now can remove all of the test cases that are shown in Listing 1, replacing them with one simple invocation:

```
should_validate_presence_of :firstname, :lastname, :email_address
```

Shoulda comes with a large number of macros that can help you test your ActiveRecord models in this way. For example, you can test all of the validations defined for the Person model using Shoulda macros:

```
should_validate_presence_of :firstname, :lastname, :email_address
should_validate_uniqueness_of :email_address
should_validate_numericality_of :grade_in_school
should_ensure_value_in_range :grade_in_school, (1..12),
 ➥:low_message => 'must be greater than or equal to 1',
 ➥:high_message => 'must be less than or equal to 12'
```

Notice how the Shoulda macros' names reflect the names of the ActiveRecord validators. This was done after Shoulda was first released, which means that some of the documentation you see on-line might be slightly out of date and include deprecated macro names.

Not surprisingly, Shoulda's authors make it possible for you to create your own macros, much as you might create your own validator method for an ActiveRecord class.

Also notice that in order to ensure that grade_in_school is numeric and that it is within a certain range, conditions that are set by a single validation line might sometimes require more than one Shoulda macro. In the particular case that I demonstrate here, there was a surprising mismatch between the error message that Rails gave to Shoulda and the message that Shoulda was expecting, in checking to see that the person's grade in school is in an acceptable range. In the end, I got around the problem by telling Shoulda what messages to expect from Rails. Although this is more verbose than I might have liked, it demonstrates the flexibility Shoulda offers.

Not surprisingly, Shoulda's authors make it possible for you to create your own macros, much as you might create your own validator method for an ActiveRecord class. I don't go into creating such macros here, but it is fairly well documented, and it means you can create a large number of tests, package them together under a single Shoulda macro and then use those tests (via the macro) across one or more projects.

Tests and Contexts

Already, you probably can see how Shoulda macros can reduce the amount of code you need to write. Shoulda also provides an RSpec-like facility that makes it possible to name tests using strings, rather than method names. Granted, this is now included in Test::Unit, albeit using a slightly different syntax. But, you can define tests using the should keyword, rather than test, which adds a bit of readability—especially when used in conjunction with contexts, which I describe below.

Here, I create a single method in the model, fullname, which returns the concatenation of the person's first and last name:

```
def fullname  # added to app/models/person.rb
  "#{firstname} #{lastname}"
end
```

Next, I add a new test:

```
should "return the concatenation of the first and last name" do
  person = Person.new(:firstname => "First",
                      :lastname => "Last",
                      :email_address => "email@example.com")

  assert_equal person.fullname, "First Last"
end
```

Now, there's nothing wrong with this test. It not only passes, but it also does a good job of checking that you are getting the right values back. Maybe it's just me, but I sometimes end up with very long lists of tests and end up categorizing them using comments inside the test file. Shoulda provides contexts that let you group tests within your file, using code rather than comments. It's obviously a bit silly to have a single context and a single test, but as with many things in the TDD/BDD world, it's worth doing things right even from the beginning, because you know that your codebase will grow over time, making it difficult to organize things correctly.

Rob Purdie
Project Manager
economist.com
amnesty.org
Lullabot-Powered
The most super powered sites in the world are created in Drupal, by you and Lullabot.
Lullabot™
New Lullabot Learning Series training DVDs at Lullabot.com

To define a context, you merely write:

```
context "Defined methods" do
    # "should" blocks go here
end
```

In other words, you now can rewrite the test block as:

```
context "Defined methods" do
    should "return the concatenation of the first and last name" do
        person = Person.new(:firstname => "First",
                            :lastname => "Last",
                            :email_address => "email@example.com")

        assert_equal person.fullname, "First Last"
    end
end
```

With a context block and a should block, you now can read your test as, "Defined methods should return the concatenation of the first and last name." It's not the most amazing description in the world, but it's not a bad start. And besides, now you can add additional should blocks to test other defined methods.

A context may contain other contexts, as well as should blocks. This means that if you have a particularly complicated model you want to test, you can have a hierarchy of contexts, with should blocks at the bottom.

Moreover, using a context block means that you can write a setup block, which defines variables and otherwise allocates resources that will be used inside a should block. You could, for instance, now write:

```
context "Defined methods" do
  setup do
    @person = Person.new(:firstname => "First",
                         :lastname => "Last",
                         :email_address => "email@example.com")

  end

  should "return the concatenation of the first and last name" do
    assert_equal @person.fullname, "#{@person.firstname}
    ➥#{@person.lastname}"
  end
end
```

As you can see, variables that are shared between a setup block and a should block need to be instance variables, their names preceded by an @ sign.

When a test is invoked, all the setup blocks within all of its surrounding contexts are invoked first. This means if a should block is within three nested contexts, and if each of those contexts has its own setup block, all three will fire before the test is executed.

Conclusion

If you are using Test::Unit to test your Ruby on Rails application, Shoulda is a natural fit, allowing you to write a large number of common tests using flexible, easy-to-read macros. In this article, I cover uses of Shoulda only for ActiveRecord models; other parts of Shoulda work with controller tests, providing additional features that can be of use for testers.

From my perspective, using Shoulda is a no-brainer. I have used it in a number of projects already and found that it further lowered the threshold to TDD/BDD, helping make my code that much more reliable. If you are new to testing, Shoulda is a great way to get started, providing an easy way to increase the stability and correctness of your code. All in all, Shoulda is a great resource for Ruby programmers in general and Rails programmers in particular.■

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

The home page for Shoulda is **thoughtbot.com/projects/shoulda**. The documentation here is a good starting point, but you probably will need to play with it a bit to get the hang of things. Even the small problem I described above, in testing the minimum and maximum ages for a person, showed that you still might need to poke through the documentation to understand things fully.

A PDF cheat sheet for Shoulda is at **kylebanker.com/assets/content/2008/shoulda_cheat_sheet.pdf**, and the popular cheat sheet program for Ruby programmers also has an entry: **cheat.errtheblog.com/s/shoulda**.

The following are a few interesting blog posts about Shoulda that also can provide some useful ideas: **pragdave.blogs.pragprog.com/pragdave/2008/04/shoulda-used-th.html**, **giantrobots.thoughtbot.com/2009/2/3/speculating-with-shoulda** and **www.alexjsharp.com/2008/10/15/shoulda-painless-unit-testing**.

LINUXCON

PORTLAND 2009

SEPTEMBER 21-23

Guiding the Linux Ecosystem

The Linux Foundation presents a brand new technical conference gathering developers, administrators and users of Linux for collaboration, advancement and interaction. Attend LinuxCon and leave more knowledgeable and better positioned for success in the year to come.

- 75 Conference Sessions, Tutorials and Mini Summits
- Developer, Operations and Business Tracks
- Education and Collaboration Opportunities
- Novell SUSE Workshop and On-Site Linux Training
- Speakers include: Linus Torvalds, Mark Shuttleworth, Bob Sutor, James Bottomley and Matt Asay.

Register Now

For more information, to become a sponsors or to register, please visit: http://events.linuxfoundation.org/events/linuxcon

Copyright © 2009 Linux Foundation. All rights reserved. Linux is a registered trademark of Linus Torvalds.

MARCEL GAGNÉ

Cross at Your Platform?

Open protocols, baby—it's the only way. If you need to carry on an instant-messaging conversation, why not do it in the privacy of your own server?

Without a doubt, I am impressed by your dedication to open standards of communication, François, but this is a little crazy—not that crazy is beyond you, *mon ami*, but you are on the verge of outdoing yourself. Smoke signals? In the restaurant? Yes, I realize it's an ancient form of communication suitable for short messages, and I am willing to let you try many things for the sake of open source and open standards, but I must say no to fires in the restaurant—unless you are making crème brûlée, of course. Besides, smoke signals require line of sight for meaningful communication. Even if I could allow it, it just won't work in the restaurant.

Quoi? What about your instant-messaging service? You promised your cousins you'd set up something? No problem, François, I've got something on tonight's menu that will do the job nicely. Now, please clean up this mess quickly. Our guests will be here momentarily. Hurry! I see them approaching now.

Welcome, *mes amis*, to *Chez Marcel*. Forgive the mess. My faithful waiter is taking care of it. In the meantime, please find your tables, sit down, and make yourselves comfortable. François, as soon as you are done, please go down to the wine cellar and bring back tonight's wine. We have a case of 2007 Jean-Max Roger Sancerre Cuvée les Caillottes Sauvignon Blanc from the Loire Valley in the south wing. This is a great medium-bodied white, *mes amis*, crisp with great citrus flavor. Enjoy!

François and I were discussing open instant messaging when you arrived. If widespread acceptance of a technology by businesses large and small constitutes a serious technology, then instant messaging is all grown up now. Although great for casual, always-on conversation, instant messaging, or IM for short, has moved firmly into the corporate network infrastructure. IM allows you to remain in contact with your fellow workers, team members and so on by carrying on short, ongoing conversations. And, it's good for family and friends as well.

Here at *Chez Marcel*, we believe strongly in open source and open protocols, and that philosophy also extends to instant messaging. If you've used any kind of instant messaging, you know there are many providers and many protocols—all of them using proprietary *standards*. There is, however, a real industry standard known as XMPP (extensible messaging and presence protocol). It's more commonly known as Jabber, and it's used by many companies and organizations. (Jabber/XMPP is the protocol used by Google Talk.)

From a business standpoint, Jabber should be your clear IM choice. Because Jabber is an open protocol, it doesn't belong to anyone in particular, so there is no single company driving its destiny. Your business won't get locked down by proprietary formats. Jabber also uses a decentralized approach, so the system is more robust. Best of all, any company can run its own private, secure, standards-compliant, Jabber instant-messaging server for little or no cost for the software. One of my favorite Jabber servers comes from a company called Jive Software. It's called Openfire, and it's completely open and released under the GPL.

Getting an Openfire Jabber server up and running starts with a visit to the Jive Software's Ignite Realtime community site at **www.igniterealtime.org**. Click on Products, then select the Openfire Jabber collaboration server link (at the time of this writing, the version number is 3.6.4). Jive and Ignite Realtime have many products listed on the site, and all of them are meant to enable collaboration and communication, but I concentrate only on Openfire here. The package comes in an RPM format package as well as DEB. There's also a tarred and gzipped bundle that should work in environments where RPM or DEB might be an issue. Installing either version of the package is easy. To install the RPM package, type the following:

```
sudo rpm -i openfire_3.6.4-1.i386.rpm
```

If you choose to use the Debian package, you can install it with:

```
sudo dpkg -i openfire_3.6.4_all.deb
```

If you need to use the tarred bundle, extract it in the /opt directory. This is the installation folder for the RPM package as well. Openfire files and programs wind up under /opt/Openfire. One plus of the RPM package is that it comes with the Java Runtime Environment (JRE). If you choose (or need to use) the tarred bundle, you also need version 1.6 Java RE loaded on your system. Java is, of course,

available from **java.sun.com**. Debian (or Ubuntu) users also need an installed Java JRE. In addition, that whole thing about everything in /opt doesn't apply to Debian users.

The installation process takes only a few seconds on modern systems. You'll see a little message that says, "Setting up Openfire" followed by a couple more messages advising you that a new user is being added (named Openfire) and that the server is starting. And, that's pretty much it. The final part of the installation, which involves configuring the server, takes place using your Web browser. The server takes only a few seconds to initialize, and the installer automatically starts the server. Of course, if this is a new install, there are a few more steps, and those are done via your Web browser. The Openfire server administrative interface runs on port 9090, so point your browser to the following address: http://localhost:9090.

A short question-and-answer setup session follows. You are asked for your preferred language—choices include French, English, German, Spanish and six others. Click Continue, and enter the server's fully qualified domain name and the port on which it operates. The 9090 port is the default, along with port 9091 for secure connections to the server. Unless you have a good reason, it probably makes sense to accept those defaults.

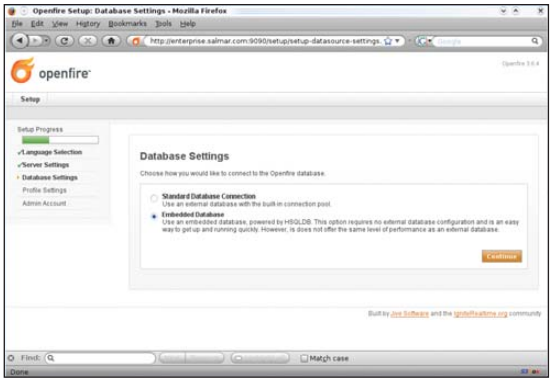


Figure 1. There are only a few steps to installation, and one of the most important is database setup.

The next screen is the database selection screen (Figure 1). Openfire supports several database

Expert included.

Meet Victoria (on the right). She is the Silicon Mechanics marketing expert responsible for the events and promotions that keep our customers informed about exciting new products and technologies. She's pictured here with her twin sister Veronica, an industrial designer, to help us make a point about what makes twin servers from Silicon Mechanics so popular. Victoria and Veronica are twins, but they don't look exactly alike and they don't do the same job. Twin servers are two servers in a single 1U chassis: they can be configured differently, and they handle their own individual workloads.

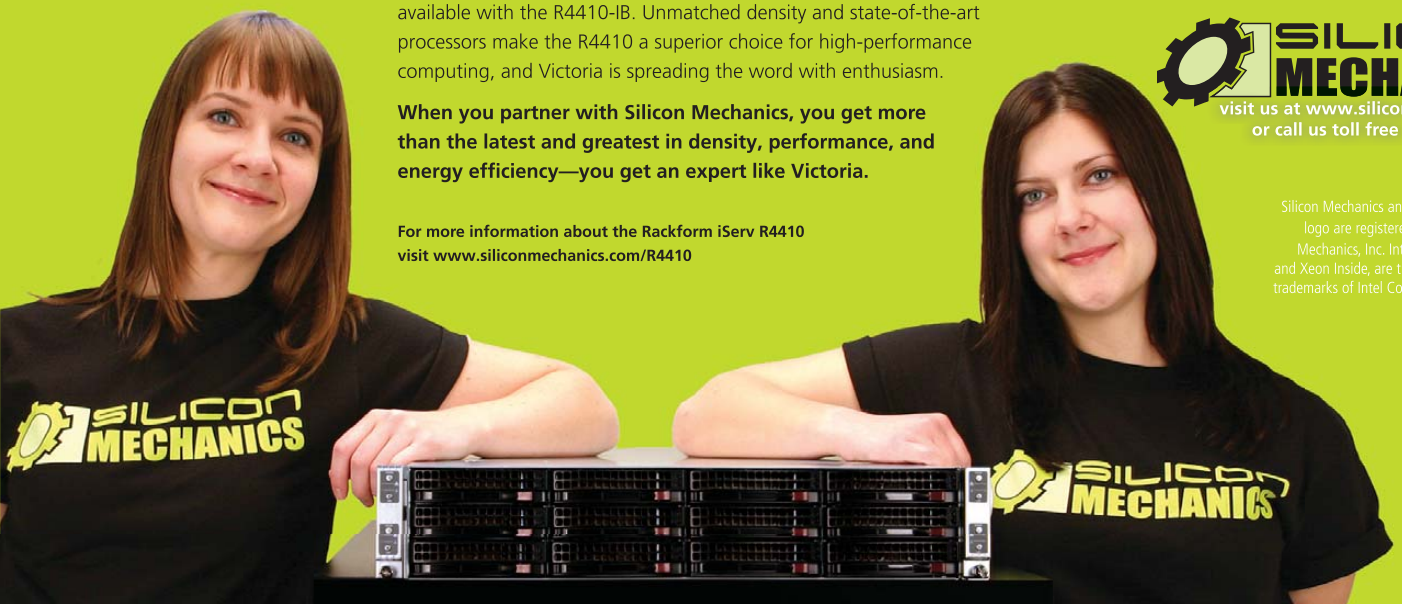
With the introduction of the Rackform iServ R4410 from Silicon Mechanics, twin power has reached a whole new level: the twin². A twin² is a 2U 4-node system. It supports four swappable, full-featured nodes in a 2U chassis with redundant power. In each node you'll find 2 of the new Intel® Xeon® 5500 Series processors, 12 DDR3 DIMM slots, 3 hot-swap drives, and an integrated dual-port GigE adapter. Integrated InfiniBand is also available with the R4410-IB. Unmatched density and state-of-the-art processors make the R4410 a superior choice for high-performance computing, and Victoria is spreading the word with enthusiasm.

intel Xeon inside™

Powerful. Intelligent.

When you partner with Silicon Mechanics, you get more than the latest and greatest in density, performance, and energy efficiency—you get an expert like Victoria.

For more information about the Rackform iServ R4410 visit www.siliconmechanics.com/R4410



SILICON MECHANICS

visit us at www.siliconmechanics.com or call us toll free at 866-352-1173

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, and Xeon Inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries.

architectures, including PostgreSQL, MySQL, Oracle and others. Each of those requires some external setup, but the documentation covers that well. If your needs are modest, select the embedded HSQLDB database included with Openfire.

For many, the built-in database will suffice and serve modest requirements well. In a larger office environment, or if you expect to have many users, you should use one of the other database options (Figure 2). Read each line carefully, because you need to enter the database name, user name and password to continue.

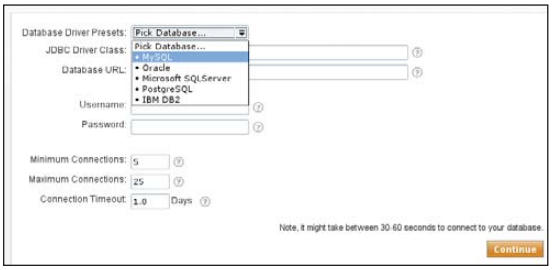


Figure 2. Several popular database formats are supported in addition to the built-in database.

Next, choose where to store your user profiles. You can select the Openfire database (the easiest choice), an existing directory server (such as an LDAP server) or Jive's Clearspace social business software. Click Continue, and it's time to set up the admin account (cleverly called admin). Provide an e-mail address for the admin user and a password, and click Continue. Congratulations! You have a running Openfire XMPP (or Jabber) server. This is the last time you will see the setup screen. From now on, when you click on the Web server address, you'll be at the Admin login screen. To go there now, click the Login to the admin console button on the page.

A quick note on procedure: if you just go ahead and click that button, you may find that you can't log in immediately via the admin console. Here's a tip. Before doing anything else, reload Openfire's configuration by typing `/etc/init.d/Openfire restart`.

At this point, you don't actually have to do anything else. Using your Jabber client of choice, you can create an account and start using the server. For instance, with Pidgin, the GNOME multiprotocol instant-messaging client, you could click Accounts to bring up the Manage Accounts dialog, and click the Add button. This brings up the Add Account window (Figure 3). From there, select XMPP from the Protocol list, choose a user name, then enter your server's domain name and select a password. Now, look at the bottom of the window shown in Figure 3. There's a check box labeled Create this new account on the server. Be sure to check that box.

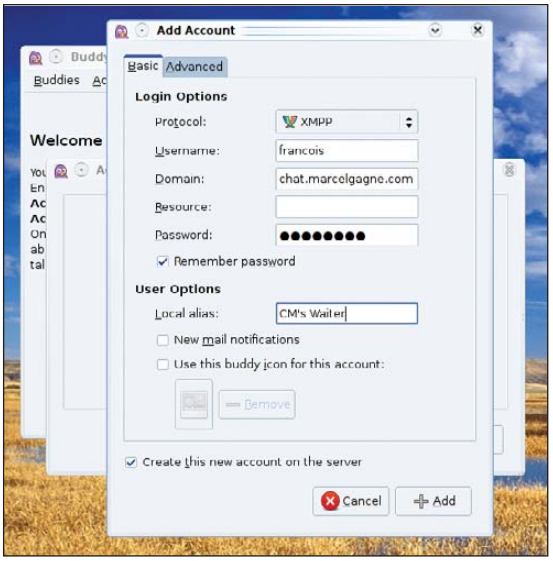


Figure 3. Setting Up a Jabber Account Using Pidgin

When you click the Add button, another window appears, and this one asks you to validate the SSL certificate from the Openfire server. Click Accept, and another window appears to confirm your registration. Enter your authentication information (user name, password and e-mail address), then click OK. The server finishes your registration, and you'll get a successful registration box. Click OK, and that's it. You'll be back at the account listing screen at this point, but not logged in, so click the enabled button, and you should be ready to start building your buddy list (Figure 4).

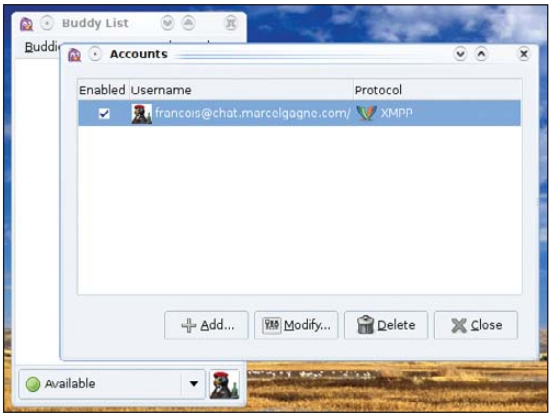


Figure 4. Logging in with Pidgin is done by enabling the account under the account manager.

Over in KDE-land, we have the Kopete multiprotocol instant-messaging client. The registration process is similar. From the main Kopete window, click Settings, then Configure. From the configuration window, select

Accounts from the left-hand sidebar, then click Add Account on the right. You'll see a window asking you to choose from one of many instant-messaging protocols. In this case, select Jabber, then click Next. This takes you to step two, the account information window from which you can register your new account. There are four tabs here, but you need to concern yourself only with the Basic Setup at this time. Enter a Jabber ID in the format of username@your.jabberserver.dom, click the Remember Password check box (assuming you don't want to enter it each time you log in), and enter a password. Now, click the Register New Account button. A Register New Jabber Account dialog appears (Figure 5).

Everything here should be filled in properly. Confirm the password, then click Register. Back at the Account Information window, click Next, and then click Finish to wrap it up. You should be logged in to your new Jabber/XMPP account automatically and ready to chat. You aren't limited to chatting with users only on this server. You can chat with any other person using Jabber IM, including people using Google Talk. Some enterprise applications even are including Jabber servers and

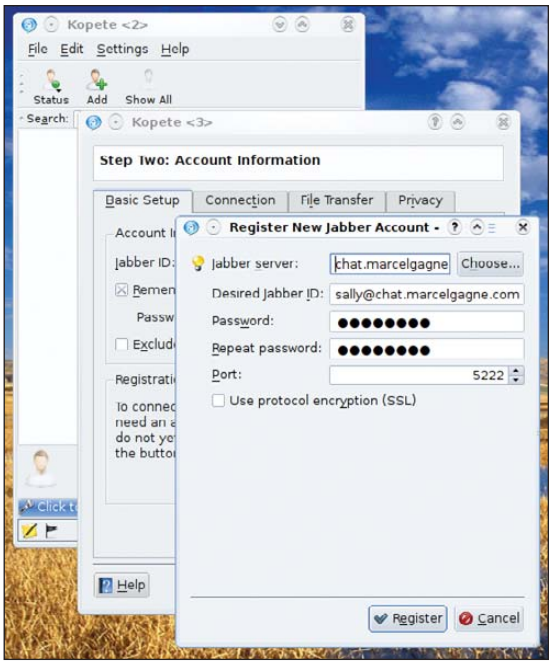


Figure 5. Setting Up a Jabber/XMPP Account with Kopete

Powerful: Rhino

Rhino M6400/E6500

- Dell Precision M6400/ Latitude E6500
- 2.2-3.0 GHz Core 2 Duo or 2.5 GHz Core 2 Quad
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3700M
- 80-500 GB hard drive
- Up to 16 GB RAM
- DVD±RW or Blu-ray
- 802.11a/g/n
- Starts at $1330

- High performance NVidia 3-D on a WUXGA widescreen
- High performance Core 2 Quad, 16 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:

Tablet: Raven

Raven X200 Tablet

- ThinkPad X200 tablet by Lenovo
- 12.1" WXGA w/ X@1280x800
- 1.2-1.86 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-320 GB hard drive / 128 GB SSD
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at $2200

Rugged: Tarantula

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 8 GB RAM
- 80-320 GB hard drive
- Call for quote

EmperorLinux

...where Linux & laptops converge

www.EmperorLinux.com

1-888-651-6686

Model specifications and availability may vary.

chat clients into their software.

By default, access is open and anyone may register. If you are running a private, corporate server, this may not be what you want to do. Securing access is done through Openfire's administrative Web interface, which provides an easy way to administer all of Openfire's functions.

Administrative tasks are organized behind a system of tabs, with functions organized into major categories. Those tabs then can be broken up into subtabs. For instance, to add users manually, click the Users/Groups tab, then select Create New User from the menu, and enter the information directly into the Web form (Figure 6). You can add, modify or delete users, organize them into groups and more. The User Summary screen makes it easy to check your users' on-line status, whether they are logged on and when they last logged out.

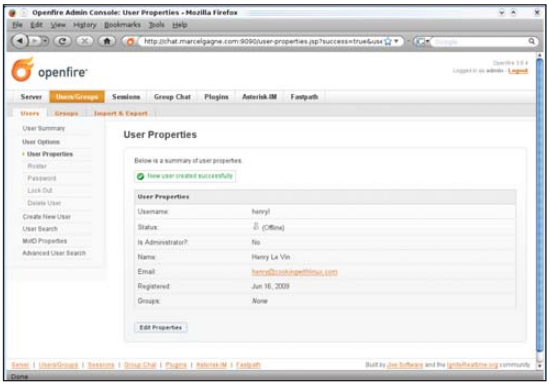


Figure 6. Creating and modifying users also can be done via the Web interface.

I started out by telling you that any and all users were allowed to register an account by default, and that's all well and good, but it may not be what you want. To control access to certain IP addresses (a local area network, for instance) or whether public registration even is allowed, click the Server tab and

The cool thing about plugins is that you can install them on the fly on the running server.

select the Server Settings subtab. Next, choose Registration & Login from the menu on the left. This page lets you configure the rules that govern user registration (Figure 7).

What seems so simple to your instant-messaging users actually is a fairly complex and exceedingly powerful collaboration server. The administrator has extensive control over Openfire's operation, from server-to-server communications, message audit

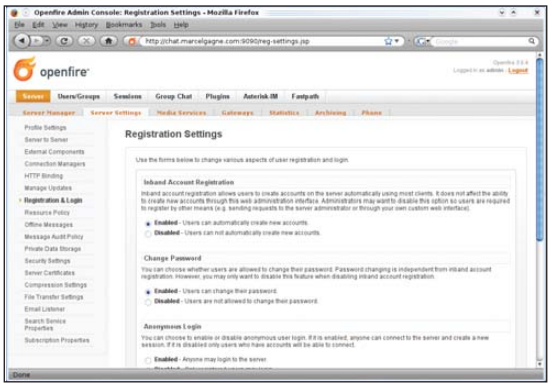


Figure 7. Define the rules by which users can (or can't) register with the system.

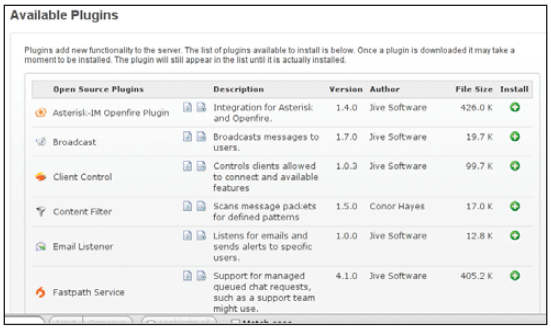


Figure 8. The plugin list will make you feel like a kid in a candy store.

policies, the treatment of messages sent to off-line users (stored, by default), private data storage, file transfers, security settings (this includes encrypted communications) and a lot more. Openfire also is extensible with added functionality provided through a system of plugins (Figure 8).

The cool thing about plugins is that you can install them on the fly on the running server. There's Asterisk VoIP integration, various filters, e-mail listeners (to alert users when new messages arrive), a live Web-based chat response system (as on customer support sites), content filters, a SIP phone plugin, monitoring extensions and lots more. To install other plugins, click the Available Plugins link to see what's available. Each plugin is listed with a description of its function, so you can decide whether it's something you need. Adding plugins also changes the administrative interface by adding new tabs—you aren't going crazy, the interface really is changing before your eyes.

Then, there are chat rooms. We all love group chats, or conferencing, if you prefer. Permanent chat rooms can be created where users can gather for general meetings or predefined functions. Rooms can be customized to define the maximum number of users,

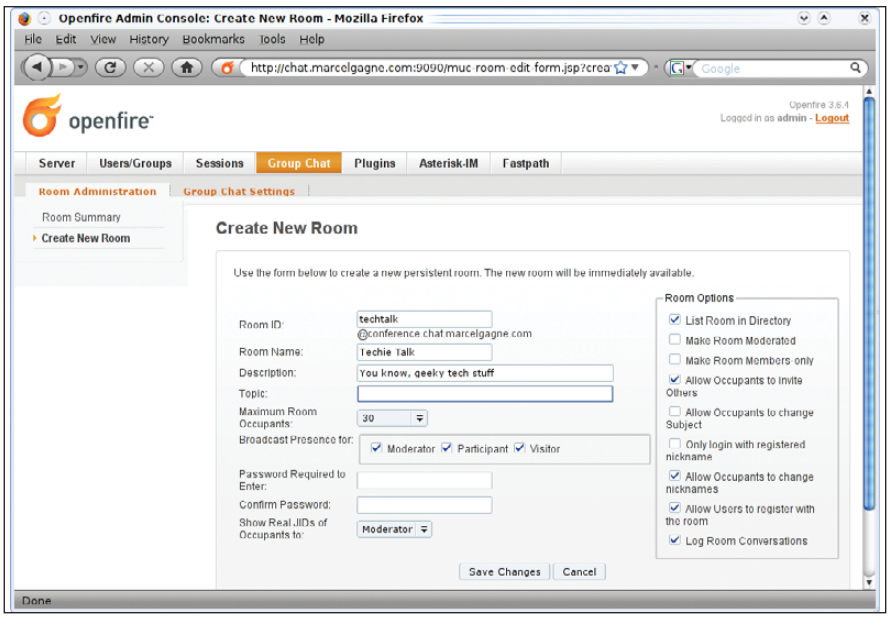


Figure 9. Openfire lets you create custom, permanent chat (or conference) rooms.

password protection, public vs. private, what users can do in the room, whether the room is moderated, amount of chat history and more (Figure 9).

You can define administrators according to their Jabber IDs, specify who can create new chat rooms, room owners, members and outcasts.

Remember, *mes amis*, free and open protocols, open standards and open source are the reasons why you should consider Jabber for your instant-messaging needs. Furthermore, with a cool, open-source product like Openfire, your company or organization's server practically is begging for you to give up that proprietary instant-messaging nonsense and move to Jabber/XMPP. Open standards and open protocols mean you aren't locked in to this or any Jabber implementation. It also means your data and your messages always will be yours, and that makes great business sense. Heck, it makes great sense, period.

Well, *mes amis*, that clock is busy reminding us that closing time is once again here. Now that you've got access to a great instant-messaging system, we can keep in touch long after we leave each other tonight. But, let's not rush our departure quite yet. There is still more wine, and I know that my faithful waiter, François, would love nothing more than to refill everyone's glass a final time before we say goodbye. Until next time, 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 marcelgagne.com and cookingwithlinux.com.

Resources

Kopete: **kopete.kde.org**

Openfire Server at Ignite Realtime: **www.igniterealtime.org/projects/openfire/index.jsp**

Pidgin: **www.pidgin.im**

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

Cooking with Linux: **cookingwithlinux.com**

WFTL Bytes!: **wftlbytes.com**

YOUR TECHNOLOGY SHOULD WORK AS FAST AND SMART AS YOU DO.

And choosing Linux should never limit your technology options.

We have more than 500 Service Providers serving more than 12 million end-users in 125 countries with our Linux-based solution.

Talk to the people who know Linux.

Talk to Parallels.

Parallels™
Optimized Computing

offer@parallels.com
425.282.6448
www.parallels.com

DAVE TAYLOR

Messing around with ImageMagick

GUIs? We don't need no stinkin' GUIs; we've got ImageMagick. Work with images from the command line.

I've written previously about working with graphic images within shell scripts, and obviously, it's a little bit tricky because, well, scripts generally are strongest working with text, and you can't even see graphics, let alone manipulate them directly. Further, let's be candid, the suite of utilities included with a stock Linux/UNIX system doesn't include much that help you work with graphics or image files at all.

Fortunately, there's a splendid open-source package called ImageMagick, which actually is designed to make working with image files from the command line easy and fast. It's the smart back end to a bunch of image utilities, and with a quick trip to **www.imagemagick.org**, you can download it too.

A couple different steps are involved in installing it, and this time, I'm actually going to play with my Apple MacBook Pro and install the utilities to live within the Darwin world of Mac OS X.

Installing ImageMagick in Darwin/NetBSD

Since 99% of the time that I'm using my Mac I am logged in as taylor, I'm going to opt to drop the software into my own personal bin directory rather than the more standard location of /usr/local/src (with the binary in /usr/local/bin). It might be that I'm a long-term UNIX geek or something, but I have my own ~/bin (or $HOME/bin, if you prefer) directory anyway, so once the binary file was downloaded, here's what I did:

```
cd ../bin
tar xvf ../Downloads/ImageMagick-i386-apple-darwin9.6.0.tar
```

Because this particular distro includes pre-compiled binaries, it's as easy as just tweaking a few environment variables to add the unpack directory and proceed:

```
export MAGICK_HOME="/Users/taylor/bin/ImageMagick-6.5.2"
export PATH="$MAGICK_HOME/bin:$PATH"
export DYLD_LIBRARY_PATH="$MAGICK_HOME/lib"
```

These are best added to your ~/.profile or ~/.cshrc (if you're using Csh, but why would you?), so that they're invoked each and every time you log in or, in the case of the Mac environment, spawn a new Terminal shell.

It's a good idea to test the newly installed programs too. Find a .gif, .jpg or .png file and see what the ImageMagick identify program says. Here's how I did that:

```
$ find . -name "*png" -o -name "*.jpg" -o -name "*gif"
./iphone-id.png
$ identify iphone-id.png
iphone-id.png PNG 470x118 470x118+0+0 8-bit DirectClass 12.2kb
```

It's more useful than the file command, which reports:

```
$ file iphone-id.png
iphone-id.png: PNG image data, 470 x 118, 8-bit/color RGB, non-interlaced
```

Where identify really shines is with JPEG files, which the file command can't quite seem to figure out. Why that's true, I don't know, but that shortcoming is the main reason I have ImageMagick installed on my system.

Doing Something Useful with ImageMagick

One of my hobbies is photography, and as a parent, I find that I frequently end up as the "official" photographer for school events. I recently did just that for my daughter's May Fair event, and I ended up with about 500 5–8MB image files that were great for printing (about 4,200x2,800) but not so good for viewing on the computer screen. What I wanted to do was create images that were approximately 1,024x800 or thereabouts, so that they'd view at 100% on a typical computer screen, in a directory that paralleled the original image file directory. That way, parents could view a slideshow of the smaller images and then grab the identically named big image if they wanted to upload it and order prints.

With ImageMagick, this is easy. In fact, if I wanted to use the mogrify command, I could have very easily done everything in a single command, but because I like obscure, complicated solutions rather than simple, elegant ones, I decided to use the convert command instead.

The challenge is that, like everything else in ImageMagick, the convert app has a staggering number of different command flags. Type `convert`, and you'll see what I mean.

Digging through them, here's the flag I want to use:

```
-resize geometry      resize the image
```

That sounds like what we need is to resize the images, though "geometry" is still a bit of an unknown. Now it's time to pop over to the ImageMagick Web site, where we find a ton of options for geometry, including:

- scale%: height and width both scaled by specified percentage.
- scale-x%xscale-y%: height and width individually scaled by specified percentages.
- width: width given, height automatically selected to preserve aspect ratio.
- xheight: height given, width automatically selected to preserve aspect ratio.
- widthxheight: maximum values of height and width given, aspect ratio preserved.

To accomplish the conversion we want, we simply can specify the desired width and let the utility do all the work:

```
$ identify DSC_7466.JPG
DSC_7466.JPG JPEG 4288x2848 4288x2848+0+0
 ➥8-bit DirectClass 8.148mb
$ convert -resize 1024 DSC_7466.JPG smaller-DSC_7466.JPG
$ identify smaller-DSC_7466.JPG
smaller-DSC_7466.JPG JPEG 1024x680 1024x680+0+0 8-bit
 ➥DirectClass 776kb
```

As hoped, the 4,288x2,848 image is shrunk down to 1,024x680, and the new, smaller image is saved with the new filename.

Great! A quick `mkdir smaller`, and we're in business, so I utilize a shell for loop to iterate through the 500 images:

```
for filename in *.png
do
  convert -resize "50%" $filename smaller/$filename
done
```

Once you've gone through the hassle of installing the ImageMagick program, it's delightful to see how easily many different tasks can be accomplished.■

Dave Taylor has been involved with UNIX since he first logged in to the on-line network in 1980. That means that, yes, he's coming up to the 30-year mark now. You can find him just about everywhere on-line, but start here: www.DaveTaylorOnline.com. In addition to all his other projects, Dave is now a film critic. You can read his reviews at www.DaveOnFilm.com.

ASA COMPUTERS

Want your business to be more productive?

The ASA Servers powered by the Intel Xeon Processor provide the quality and dependability to keep up with your growing business.

Hardware Systems for the Open Source Community - Since 1989.

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.

1U Server - ASA1401i

- 1TB Storage Installed. Max – 3TB.
- Intel Dual core 5030 CPU (Qty-1), Max-2 CPUs
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 4X250GB htswap SATA-II Drives Installed.
- 4 port SATA-II RAID controller.
- 2X10/100/1000 LAN onboard.

2U Server - ASA2121i

- 4TB Storage Installed. Max – 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

3U Server - ASA3161i

- 4TB Storage Installed. Max – 12TB.
- Intel Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives Installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

5U Server - ASA5241i

- 6TB Storage Installed. Max – 18TB.
- Intel Dual core 5050 CPU,
- 4GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 24X250GB htswap SATA-II Drives Installed.
- 24 port SATA-II RAID. CARD/BBU.
- 2X10/100/1000 LAN onboard.
- 930w Red PS.

8U Server - ASA8421i

- 10TB Storage Installed. Max – 30TB.
- Intel Dual core 5050 CPU.
- Quantity 42 Installed.
- 1GB 667MGZ FBDIMMs.
- Supports 32GB FBDIMM.
- 40X250GB htswap SATA-II Drives Installed.
- 2X12 Port SATA-II Multilane RAID controller.
- 1X16 Port SATA-II Multilane RAID controller.
- 2X10/100/1000 LAN onboard.
- 1300 W Red Ps.

All systems installed and tested with user's choice of Linux distribution (free). ASA Collocation—$75 per month

2354 Calle Del Mundo,
Santa Clara, CA 95054
www.asacomputers.com
Email: sales@asacomputers.com
P: 1-800-REAL-PCS | FAX: 408-654-2910

Intel®, Intel® Xeon™, Intel Inside®, Intel® Itanium® and the Intel Inside® logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Prices and availability subject to change without notice. Not responsible for typographic errors.

intel Xeon® inside™

Powerful. Efficient.

AppArmor in Ubuntu 9

Psst! Your Ubuntu system has been secretly hardened with AppArmor!

MICK BAUER

Three years ago, I devoted a couple columns (in the April and August 2006 issues of *LJ*) to Novell AppArmor, a partial implementation of Mandatory Access Controls (MACs) that Novell had integrated into SUSE Linux as part of its acquisition of Immunix. Novell also had released AppArmor's source code under the GPL. I expressed hope that other distributions soon would offer AppArmor as an easier-to-configure alternative to SELinux.

The good news is, during the three years since I wrote those articles, both Ubuntu and Mandriva have incorporated AppArmor into their respective distributions. Although until recently Ubuntu hasn't provided very much documentation on its AppArmor port—one might even characterize Ubuntu's AppArmor adoption as stealthful—AppArmor actually has been in Ubuntu since Ubuntu 7.10 (Gutsy Gibbon). In fact, I mentioned this inclusion in these very pages in the April 2008 issue, in my article "Security Features in Ubuntu Server".

At the time, I commented that due to its lack of AppArmor GUI tools or documentation, AppArmor in Ubuntu 7.10 appeared to be targeted at expert users. With Ubuntu 9.04, I'm happy to report that although AppArmor in Ubuntu still is configured strictly via the command line, it's now amply documented and comes with a useful set of default profiles.

The bad news is, in late 2007, Novell laid off all four of its full-time AppArmor developers, raising serious questions about the future of AppArmor (see The Future of AppArmor sidebar).

Being a security goon, I'm not optimistic by nature. However, I do believe in making hay while the sun shines. If a compelling tool is available to you in Ubuntu 9.04, you should take advantage of it and not worry about whether that tool will be available in Ubuntu 11.04—unless, of course, that tool requires an enormous investment in your time, attention and thought.

But AppArmor, unlike most other MAC mechanism, is not such a tool. As I explain in this month's overview of AppArmor in Ubuntu, for many applications, you don't need to do *anything* to enable or configure AppArmor protection. For others, AppArmor essentially can train *itself* in protecting them.

So, let's take a look at AppArmor in Ubuntu.

AppArmor Review

In case you missed my earlier articles on this topic, AppArmor is based on the Linux Security Modules (LSMs), as is SELinux. AppArmor, however, provides only a subset of the controls SELinux provides. Whereas SELinux has methods for Type Enforcement (TE), Role-Based Access Controls (RBACs) and Multi-Level Security (MLS), AppArmor provides only a form of Type Enforcement.

Type Enforcement involves confining a given application to a specific set of actions, such as writing to Internet network sockets, reading a specific file and so forth. RBAC involves restricting user activity based on the defined role, and MLS involves limiting access to a given resource based on its data classification (or label).

By focusing on Type Enforcement, AppArmor provides protection against, arguably, the most common Linux attack scenario—the possibility of an attacker exploiting vulnerabilities in a given application that allows the attacker to perform activities not intended by the application's developer or administrator. By creating a baseline of expected application behavior and blocking all activity that falls outside that baseline, AppArmor (potentially) can mitigate even zero-day (unpatched) software vulnerabilities.

What AppArmor *cannot* do, however, is prevent abuse of an application's *intended* functionality. For example, the Secure Shell dæmon, SSHD, is designed to grant shell access to remote users. If an attacker figures out how to break SSHD's authentication by, for example, entering just the right sort of gibberish in the user name field of an SSH login session, resulting in SSHD giving the attacker a remote shell as some authorized user, AppArmor may very well allow the attack to proceed, as the attack's outcome is perfectly consistent with what SSHD would be expected to do after successful login.

If, on the other hand, an attacker figured out how to make the CUPS print services dæmon add a line to /etc/passwd that effectively creates a new user account, AppArmor *could* prevent that attack from succeeding, because CUPS has no reason to be able to write to the file /etc/passwd.

AppArmor on Ubuntu

In SUSE's and Ubuntu's AppArmor implementations, AppArmor comes with an assortment of pretested profiles for popular server and client applications

ABERDEEN
SERVERS AND STORAGE

HOW MUCH VMWARE DO YOU NEED?

VMware Certified NAS and Servers

The scalable AberNAS is the only Network Attached Storage appliance in its class to carry the VMware® Ready Certification. Aberdeen enables VMware-centric organizations of any size to affordably take advantage of AberNAS's robust features such as; superior RAID controller speed, increased scalability and optional NAS-to-NAS mirroring.

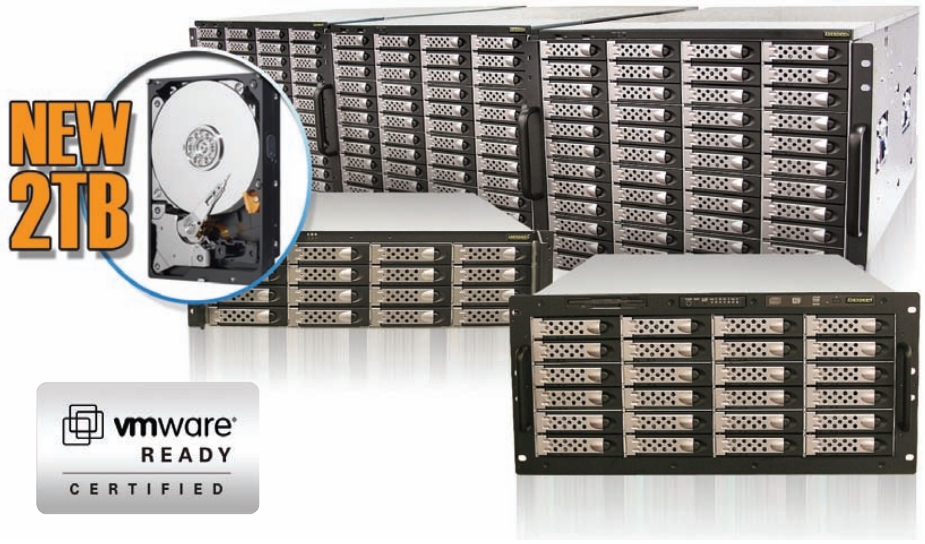
VMware Certified Aberdeen Stirling servers offer a lower cost of ownership, while eliminating concerns of compatibility complications. Stirling servers possess exceptional benefits designed to deliver improved operational efficiency within a dynamic VMware environment.

ABERNAS

VMware Certified NAS Up To 100TB

- Quad-Core Intel® Xeon® Processor 5500 Series
- 3GB/6GB 1333MHz DDR3 memory
- Supports both SAS & SATA Storage Drives
- RAID 0, 1, 5, 6, 10, 50, 60 Capable
- SAS Expansion Ports & iSCSI Target Capable
- Dual/Quad Gigabit Ethernet Ports
- **5-Year Warranty**

1U - 8TB NAS Starting at $4,995
2U - 24TB NAS Starting at..... $10,495
3U - 32TB NAS Starting at..... $12,995
5U - 48TB NAS Starting at..... $17,995
6U - 64TB NAS Starting at..... $23,495
8U - 100TB NAS Starting at..... $33,995



intel Xeon inside™
Powerful. Intelligent.

STIRLING 123

1U Dual Xeon Server

- Up to 2x Quad-Core Intel® Xeon® Processor 5400 Series
- 32GB FBDIMM max memory
- Up to 4 SATA or SAS drives
- 650W redundant power
- Pre-installed VMware® ESXi on Disk-on-Module
- **5-Year Warranty**

Starting at $1,845

STIRLING 132T

1U Twin Node Server

- Up to 2x Quad-Core Intel Xeon Processor 5400 Series
- 64GB FBDIMM max memory
- Up to 2 SATA drives
- 980W high efficiency power
- Pre-installed VMware ESXi on Disk-on-Module
- **5-Year Warranty**

Starting at $2,675

STIRLING 209

2U Xeon Server

- Single Quad-Core Intel Xeon Processor 5500 Series
- 24GB DDR3 max memory
- Up to 6 SATA drives
- 400W high efficiency power
- Pre-installed VMware ESXi on Disk-on-Module
- **5-Year Warranty**

Starting at $1,495

STIRLING 244

2U Quad Xeon Server

- Up to 4x Six-Core Intel Xeon Processor 7300 Series
- 192GB FBDIMM max memory
- Up to 6 SATA or SAS drives
- 1200W redundant power
- Pre-installed VMware ESXi on Disk-on-Module
- **5-Year Warranty**

Starting at $5,275

STIRLING 269

2U Dual Xeon Server

- Up to 2x Quad-Core Intel Xeon Processor 5500 Series
- 144GB DDR3 max memory
- Up to 8 SATA or SAS drives
- 720W redundant power
- Pre-installed VMware ESXi on Disk-on-Module
- **5-Year Warranty**

Starting at $2,095

STIRLING 444

4U Quad Xeon Server

- Up to 4x Six-Core Intel Xeon Processor 7300 Series
- 192GB FBDIMM max memory
- Up to 5 SATA or SAS drives
- 1200W redundant power
- Pre-installed VMware ESXi on Disk-on-Module
- **5-Year Warranty**

Starting at $5,345

Intel, Intel Logo, Intel Inside, Intel Inside Logo, Pentium, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. For terms and conditions, please see www.aberdeeninc.com/abpoly/abterms.htm. lj031

888-297-7409
www.aberdeeninc.com/lj031

The Future of AppArmor

AppArmor has been adopted as the default Mandatory Access Control solution for both the Ubuntu and Mandriva distributions. I've sung its praises before, and as evidenced by writing my now third column about it, clearly I'm still a fan.

But, you should know that AppArmor's future is uncertain. In late 2007, Novell laid off its full-time AppArmor developers, including project founder Crispin Cowan (who subsequently joined Microsoft).

Thus, Novell's commitment to AppArmor is open to question. It doesn't help that the AppArmor Development Roadmap on Novell's Web site hasn't been updated since 2006, or that Novell hasn't released a new version of AppArmor since 2.3 Beta 1 in July 2008, nearly a year ago at the time of this writing.

But, AppArmor's source code is GPL'd: with any luck, this apparent slack in AppArmor leadership soon will be taken up by some other concerned party—for example, Ubuntu and Mandriva developers. By incorporating AppArmor into their respective distributions, the Ubuntu and Mandriva teams have both committed to at least *patching* AppArmor against the inevitable bugs that come to light in any major software package.

Given this murky future, is it worth the trouble to use AppArmor? My answer is an emphatic yes, for a very simple reason: AppArmor is so easy to use—requiring no effort for packages already having distribution-provided profiles and minimal effort to create new profiles—that there's no reason not to take advantage of it for however long it remains an officially supported part of your SUSE, Ubuntu or Mandriva system.

and with simple tools for creating your own AppArmor profiles. On Ubuntu systems, most of the pretested profiles are enabled by default. There's nothing you need to do to install or enable them.

Other Ubuntu AppArmor profiles are installed, but set to run in complain mode, in which AppArmor only logs unexpected application behavior to /var/log/messages rather than both blocking and logging it. You either can leave them that way, if you're satisfied with just using AppArmor as a watchdog for those applications (in which case, you should keep an eye on /var/log/messages), or you can switch them to enforce mode yourself, although, of course, you should test thoroughly first.

Still other profiles are provided by Ubuntu's optional apparmor-profiles package. Whereas ideally a given AppArmor profile should be incorporated into its target application's package, for now at least, apparmor-profiles is sort of a catchall for emerging and not-quite-stable profiles that, for whatever reason, aren't appropriate to bundle with their corresponding packages.

Active AppArmor profiles reside in

Table 1. Ubuntu Packages Having AppArmor Profiles

Ubuntu Package Name	AppArmor Default Mode	Package Description
bind	enforce	The BIND DNS server
clamd	enforce	ClamAV antivirus scanner
cups	enforce	Print services dæmon
dhcp3-client	enforce	ISC's DHCP client
dhcp3-server	enforce	ISC's DHCP server
mysql	enforce	MySQL database engine
slapd	enforce	OpenLDAP LDAP server
tcpdump	enforce	Command-line network sniffer

Table 2. Packages Whose AppArmor Profiles Are Provided by apparmor-profiles

Ubuntu Package Name	AppArmor Default Mode	Package Description
ping	complain	Network diagnostic tool
klogd	complain	Kernel message logger
syslogd	complain	Berkeley system message logger
syslog-ng	complain	Syslog-NG system message logger
avahi-daemon	enforce	Multicast-DNS (network auto-discover)
dnsmasq	complain	DNS/DHCP forwarder used for Internet connection sharing
identd	complain	Maps user names to processes/sockets
mdnsd	complain	Scans for Multicast-DNS services
nmbd	complain	Part of Samba (MS file sharing)
nscd	complain	Nameservice (DNS) Caching Dæmon
ntpd	complain	Network Time Protocol Dæmon
smbd	complain	Part of Samba (MS file sharing)
traceroute	complain	Network diagnostic tool

/etc/apparmor.d. The files at the root of this directory are parsed and loaded at boot time automatically. The apparmor-profiles package installs some of its profiles there, but puts experimental profiles in /usr/share/doc/apparmor-profiles/extras.

The Ubuntu 9.04 packages shown in Table 1 put corresponding profiles into /etc/apparmor.d.

If you install the package apparmor-profiles, you'll additionally get default protection for the packages shown in Table 2.

The lists in Tables 1 and 2 are perhaps as notable for what they lack as for what they include. Although such high-profile server applications as BIND, MySQL, Samba, NTPD and CUPS are represented, very notably absent are Apache, Postfix, Sendmail, Squid and SSHD. And, what about important client-side network tools like Firefox, Skype, Evolution, Acrobat and Opera?

Profiles for those applications and many more are provided by apparmor-profiles in /usr/share/doc/apparmor-profiles/extras, but because they reside there rather than /etc/apparmor.d, they're effectively disabled. These profiles are disabled either because they haven't yet been updated to work with the latest version of whatever package they protect or because they don't yet provide enough protection relative to the Ubuntu AppArmor team's concerns about their stability.

Testing and tweaking such profiles is beyond the scope of this article, but suffice it to say, it involves the logprof command.

Creating AppArmor Profiles

At a high level, creating a new AppArmor profile involves creating a deny all policy and then running that profile in complain (log-only) mode; running your application in as typical a fashion as possible; using the resulting log messages to loosen up the profile enough (but *only* enough) for the application to work properly; and setting the finished, tuned profile to enforce mode.

AppArmor, through its genprof and logprof commands, walks you through this entire process interactively. I'm not going to cover the process for tweaking existing AppArmor profiles with logprof. logprof sessions are very similar to genprof sessions, so if you're comfortable creating new profiles, it's easy to tweak existing ones. (See Resources for

Expert included.

Carlos knows a good thing when he sees it, but as one of the Silicon Mechanics Quality Assurance Experts, that's his job. When it comes to the new Storform iServ R515, Carlos sees more than just the care and attention to detail that go into the production of your custom-configured, versatile storage server.

The new R515 from Silicon Mechanics is a 4U server with 24 hot-swap SAS/SATA drives, optional integrated SAS controller, 6 PCI expansion slots, and redundant power. It features the new Intel® Xeon® processor 5500 series with Intel Turbo Boost Technology that delivers additional performance automatically when you need it. It also offers smart features like energy-saving DDR3 memory. With 2 internal SATA drive bays, you may want to think about incorporating Intel X25-E Extreme Solid State Drives for fast, reliable, energy-efficient OS drives. And Carlos will make sure that the servers we build and ship to you are exactly what you ordered.

intel Xeon inside™

Powerful. Intelligent.

When you partner with Silicon Mechanics, you get more than cost-effective, flexible, high-capacity storage—you get an expert like Carlos.

For more information about the Storform iServ R515 visit www.siliconmechanics.com/R515

SILICON MECHANICS

visit us at www.siliconmechanics.com or call us toll free at 866-352-1173

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, and Xeon Inside, are trademarks or registered trademarks of Intel Corporation in the US and other countries.

Listing 1. A Shell Script Needing AppArmor Protection

```
#!/bin/sh
#
# spaztacle.sh : archives /var/spaetzle to specified tar-file

tar -cf $1 /var/spaetzle
```

more information on the latter.)

So, let's walk through the process of creating a new AppArmor profile. For this example scenario, let's start with a simple shell script, spaztacle.sh, that could use some protection. Listing 1 shows the script itself.

As you can see, this script allows users to create a backup archive of the directory /var/spaetzle, using the archive filename specified in the command line (for example, `spaztacle.sh mybackup.tar`). To create an AppArmor profile for it, run the following command:

```
bash-$ sudo genprof spaztacle.sh
```

What follows is an interactive question-and-answer session in which:

1. genprof creates a new AppArmor profile for spaztacle.sh, containing a simple "deny all access" policy.

2. genprof loads the new policy in complain mode and prompts you to start the application in a separate window (this is your first opportunity to demonstrate normal application activity to genprof).

3. After you've demonstrated the application sufficiently, genprof analyzes the messages the new profile generated in /var/log/messages.

4. For each log message, genprof asks what sort of rule to add to your new AppArmor profile to account for the behavior that was logged.

5. After all log messages have been analyzed, genprof allows you to repeat the test/analyze cycle, which may or may not result in additional rules for the profile.

6. When you're done with the testing/log-analyzing cycle, genprof saves the profile and loads it in enforce mode. You're done!

A full genprof session is too lengthy to list and dissect here, but we can discuss some highlights from my `sudo genprof spaztacle.sh` session that illustrate how the process works.

First, I'm asked whether genprof should query the AppArmor profile repository at opensuse.org. I select `d` to disable repository access.

Next, I'm prompted to run my application. So I open another xterm window, and from my home directory, run the command `spaztacle.sh arf.tar`. That command results in the file arf.tar being written in my home directory, as expected.

Back in the genprof session, I type `s` to begin scanning the system log for AppArmor messages. genprof asks me whether and how to allow /bin/tar to be executed. This, of course, is the core function of spaztacle.sh, so I type `i` to cause tar to be allowed, "inheriting" the same profile as spaztacle.sh itself.

Next, I'm asked whether to allow /bin/dash to run. Because spaztacle.sh is a Bourne shell script, it needs to be interpreted by /bin/dash (on Ubuntu 9.04 /bin/sh actually is a symbolic link to /bin/dash). I type `a` to allow /bin/dash to run.

Then, I'm asked whether spaztacle.sh may read itself—that is, /usr/bin/spaztacle.sh. This is an expected part of the script-parsing process; I type `a`.

For now, there are no further log messages to process, so genprof prompts me to save the tweaked profile and asks whether to scan for more events. Before answering, I switch to my other xterm window, change my working directory to /home/mick/Public, and run the command `spaztacle.sh anothertar.tar`.

Sure enough, back in the genprof session after I type `s` again, there's a new set of "complaints" to process. The first concerns whether spaztacle.sh (actually tar) can read /etc/group. I'm given the option of allowing access only to /etc/group or of enabling the abstraction called nameservice.

Abstractions are groups of commonly accessed profile objects that constitute common system functions and services, such as checking file permissions, looking up hostnames and so forth. In this case, I select the nameservice abstraction and type `a` to allow it.

Next, genprof asks me whether to allow only write access to the (new) file anothertar.tar, or to use some sort of wild card ("glob" in AppArmor parlance). Because I want users to be able to create arbitrary tar archives in their respective home directories, I type `n` to specify a new glob, and specify `/home/**`.

In AppArmor profiles, ** is a wild card that means "any string, including /". This is in contrast to *, which means "any string up to and excluding a / and anything after it". Therefore, `/home/**` means "everything within /home/, including all subdirectories of its subdirectories".

Listing 2. The Finished Profile

```
# Last Modified: Mon Jun 15 21:29:38 2009
#include <tunables/global>

/usr/bin/spaztacle.sh {
  #include <abstractions/base>
  #include <abstractions/nameservice>

  /bin/dash rix,
  /bin/tar rix,
  owner /home/** a,
  /usr/bin/spaztacle.sh r,
  /var/spaetzle/ r,
  /var/spaetzle/** r,

}
```

This implies that users might be able to write files to other users' home directories, but AppArmor controls *augment* normal Linux filesystem permissions; they don't *replace* them. In our example, therefore, users will be able to write to other other users' directories only if those directories' permissions are set accordingly.

In fact, our `/home/**` rule actually *reduces* the number of places spaztacle.sh can create tar archives. Without this rule, spaztacle.sh can write in *any* directory in which the user executing it has write privileges, not just subdirectories of /home/.

There are just two more log entries to account for. One concerns read access to /var/spaetzle. I type a to allow this access. You might be tempted to create a new glob instead, `/var/spaetzle/**`, but as it happens, tar handles the directory itself and its contents separately.

Therefore, only *after* creating the rule allowing access to /var/spaetzle and being prompted for a decision on allowing access to the file /var/spaetzle/arf.txt, will I type `n`, create the new glob `/var/spaetzle/**` and allow access to it.

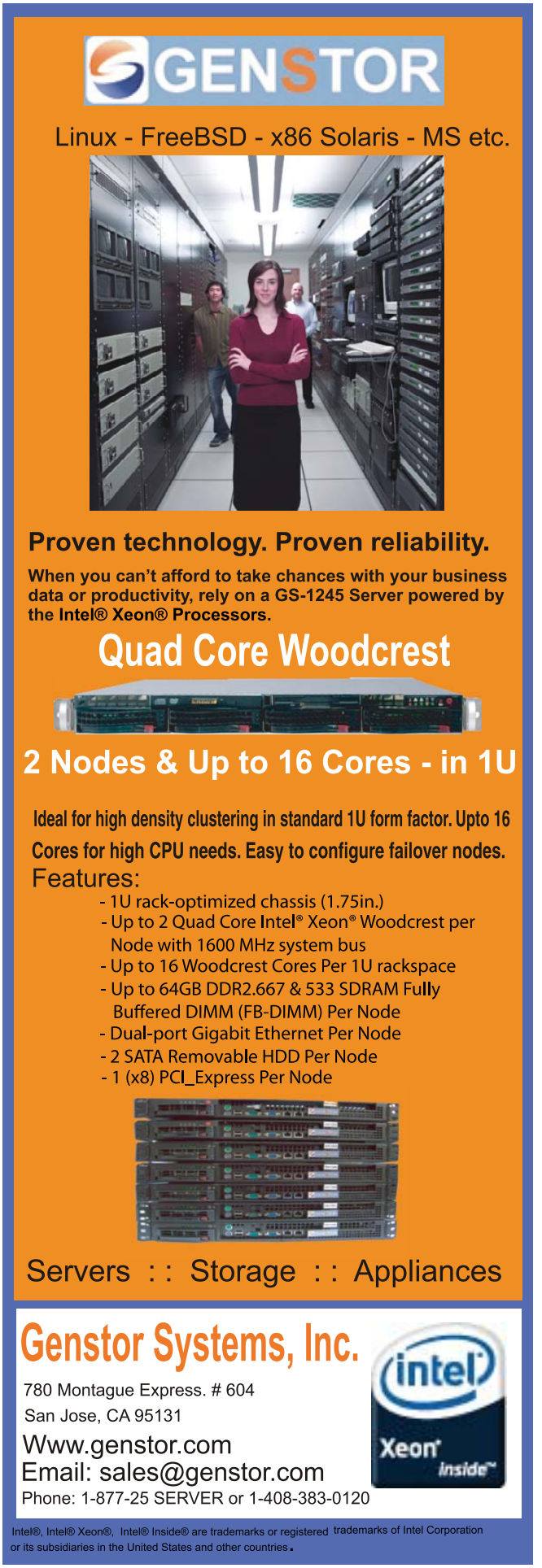
Finally, we've reached the end of the new AppArmor events in /var/log/messages. When genprof asks me what to do after saving the changed profile, I finish the genprof session. genprof puts my new profile into enforce mode, reloads it and I'm done! Listing 2 shows the result, /etc/apparmor.d/usr.bin.spaztacle.sh.

Happily, if I run spaztacle.sh again, it still works. But, is AppArmor doing anything? I can make sure the new profile is loaded with this command:

```
bash-$ sudo aa-status
```

Here's part of its output:

```
apparmor module is loaded.
26 profiles are loaded.
13 profiles are in enforce mode.
   /usr/sbin/clamd
```

```
    /usr/sbin/cupsd
    /usr/bin/spaztacle.sh
[...]
```

Great! The spaztacle.sh profile is loaded in enforce mode. Besides showing what profiles are loaded and in what mode, aa-status also lists which processes are being protected actively. Because spaztacle.sh isn't actually running at the moment, it doesn't turn up in aa-status' output as an active process, but that's okay—normally you'd expect server dæmons, not commands per se, to turn up in that part of aa-status' output.

There's just one more test we'll do to see if AppArmor is doing its job. The more astute among you may have noticed that there's a glaring flaw in my little shell script (Listing 1). Because I didn't contain $1 in quotation marks, it's possible for a mischievous user to execute spaztacle.sh like this:

```
bash-$ spaztacle.sh "tarfile.tar /etc/apparmor.d/"
```

When the tar command in spaztacle expands the command input, it will correctly interpret tarfile.tar as the target file, but will include not only /var/spaetzle but also /etc/apparmor.d/ in the tar archive! On the one hand, local file permissions still apply. This works only if users in question have read access to /etc/apparmor.d, which means that although they're tricking spaztacle.sh, they aren't copying anything they'd otherwise be unable to get at.

But on the other hand, this is *unexpected* behavior for my unfortunate script. I don't want users to be able to include arbitrary directories in their authorized backups of /var/spaetzle.

So I'm glad to see that if I actually try running spaztacle.sh that way with my new AppArmor profile in enforce mode, this is the result:

```
tar: /etc/apparmor.d: Cannot open: Permission denied
tar: Error exit delayed from previous errors
```

The following message also has been written to /var/log/messages:

```
Jun 16 01:17:43 micksbox kernel: [57354.414567] type=1503
audit(1245133063.520:1004): operation="inode_permission"
requested_mask="::r" denied_mask="::r"
fsuid=1000 name="/etc/apparmor.d/"
pid=28019 profile="/usr/bin/spaztacle.sh"
```

Success! AppArmor has correctly identified bad behavior on spaztacle.sh's part. And, the intended tar file (tarfile.tar) not only was created, it also contains the backup of /var/spaetzle that I *did* want the user to be able to create—only the unexpected part of spaztacle.sh's activity was blocked. Success indeed!

Conclusion

Using genprof may seem a little involved, but the man pages for genprof, logprof and apparmor.d explain most of what you need to know. The tutorials listed in Resources should be helpful too. I hope I've covered enough here to get you started using AppArmor on your own Ubuntu system!■

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

bodhi.zazen's "Introduction to AppArmor" for Ubuntu: **ubuntuforums.org/showthread.php?t=1008906**

Official Ubuntu AppArmor User Guide: **https://help.ubuntu.com/9.04/serverguide/C/apparmor.html**

Official Ubuntu AppArmor Overview: **www.ubuntu.com/products/whatisubuntu/serveredition/features/apparmor**

Ubuntu Community AppArmor Documentation: **https://help.ubuntu.com/community/AppArmor**

"AppArmor Is Dead" (Blog Post by Russell Coker): **etbe.coker.com.au/2008/08/23/apparmor-is-dead**

"Go Ahead, Make My Day" (Response to Coker by Crispin Cowan): **blogs.msdn.com/crispincowan/archive/2008/09/02/go-ahead-make-my-day.aspx**

Novell AppArmor Developer Roadmap: **developer.novell.com/wiki/index.php/Apparmor_dev**

Miscellaneous, Interesting AppArmor Notes on the Ubuntu Wiki: **https://wiki.ubuntu.com/AppArmor**

The OpenSUSE Project's AppArmor Page: **en.opensuse.org/Apparmor**

"Security Features in SUSE 10.0" by Mick Bauer, *LJ* April 2006: **www.linuxjournal.com/article/8783**

"An Introduction to Novell AppArmor" by Mick Bauer, *LJ* August 2006: **www.linuxjournal.com/article/9036**

"Security Features in Ubuntu Server" by Mick Bauer, *LJ* April 2008: **www.linuxjournal.com/article/10012**

However you get there, get there

Because some events are just too important to miss

Cybersecurity threats are recession-proof. Even in challenging times you can devise ways to do more with less to defend your organization against increasingly sophisticated attacks. We will help you achieve this goal at SC World Congress 2009.

You'll develop cost-effective strategies to solve cybersecurity problems while immersing yourself in the latest intelligence and practical tactics delivered by leading independent experts. Experience the newest solutions, free keynotes and a host of other activities.

Incomparable security education across four dedicated tracks

Independent expert speakers with global perspectives

A packed expo floor showcasing the best information security tools

Security Innovators Throwdown – to find the industry's most innovative new companies

Opportunities to network with your peers and earn continuing education credits from (ISC)²

SC WorldCongress
ENTERPRISE DATA SECURITY 2009
CONFERENCE & EXPO

October 13-14 at the Sheraton New York Hotel & Towers in New York City

REGISTER NOW to attend the expo and keynote sessions for free or register for the full conference pass and, for a limited time, qualify for a 33 percent early bird discount.

Visit www.scworldcongress.com.

What Really IRCs Me: Twitter

KYLE RANKIN

In my never-ending search to do all communications through the same IRC client, this month I present tircd—a great way to connect to Twitter over IRC.

In last month's column, I talked about the fact that I thought IRC was the ideal interface for quick communication with my friends. I keep an IRC session running at all times within a screen session, so I can continuously lurk in all of my channels. Because many of my friends use IM instead of IRC though, I've had to figure out ways to manage all of my communication without having a ton of different programs open. Last month, I discussed how I used Bitlbee so I could access all sorts of IM services from my IRC client, and I promised that in the follow-up column, I would talk about how to do something similar for Twitter.

A Quick Twitter Rant

In case you didn't read last month's Point/Counterpoint column, let me summarize my opinion here. I don't see the point of Twitter. I think everything people use Twitter for already could be achieved with IRC and instant messaging without the character limits. Again, IRC is my ideal way to communicate, but now some of my friends (ahem Bill) talk more on Twitter than they do on IRC. So after much prodding from Bill, I bit the bullet and registered a Twitter account so I could see what the fuss was about.

Now, just because I had a Twitter account didn't mean I was going to flood the Internet with every meal and traffic jam in my life. My main requirement for setting up the account at all was that I could access everything via IRC. That way, Twitter was nothing more than another IRC channel, only with higher latency and lower stability. To be honest, I mostly use it in "read-only" mode and just read other people's tweets.

tircd to the Rescue

It turns out I'm not the only one who wanted to access Twitter over IRC, and in fact, quite a few different programs out there provide a local IRC gateway to Twitter. Unfortunately, none of the programs have been packaged for my distribution yet, so after struggling to get a few running, I finally found one with a reasonably simple install that worked: tircd.

tircd is a simple Perl script that works much like Bitlbee. When you start the program, it creates a new IRC server on your local machine that you can connect to with an IRC client. The only difference is that it interfaces with your Twitter account, so people you follow show up as users in the channel, and their tweets show up as normal chat messages. Once you are in the channel, everything you type becomes a new Twitter message as well, so it behaves much like any other IRC channel.

To install tircd, first go to the main project page at **code.google.com/p/tircd**, and download the latest version. As with many Perl scripts, tircd makes use of some CPAN modules you might not have on your system, so dust off your Perl programmer hat, and type the following command as root to install the CPAN modules:

```
# cpan -i POE POE::Filter::IRCD Net::Twitter
```

If this is the first time you've used CPAN on your system, you first will have to go through a basic CPAN configuration process, so it knows which mirrors to use and whether you have any proxies in place. Apart from when you choose the mirrors, the default settings should be fine, and when you select the mirrors, simply pick a few that might be close by.

Once the CPAN modules are installed, extract the tircd package in some directory (your home directory works), and then change to that directory. You'll see that only a few files are inside:

```
$ tar xzvf tircd_v0.7.tar.gz
$ cd tircd
$ ls
ARTISTIC  GPL  tircd.cfg  tircd.pl  tircd.pod
```

tircd includes a sample configuration file that is heavily commented, so you can see what each option does. The default settings should work in most situations, unless you already run a local IRC server (such as Bitlbee in my case). If you do run another IRC server, change the port setting in the file from port 6667 to port 6668 so it won't conflict.

Start and Connect to tircd

To start tircd, simply execute the tircd.pl script and pass the path to the tircd.cfg file as an argument. Because I was already in the tircd directory, I could type:

```
$ ./tircd.pl ./tircd.cfg
```

You should see a number of log messages scroll by in the terminal, and if you want to use the terminal for something else, don't forget to add the & character at the end to start this in the background. Now you might be wondering how you enter your account information. All of this is set when you connect to the local server. Just use your Twitter user name and password as authentication. For instance, on most command-line IRC clients, you would type:

```
/server localhost 6667 twitter_password twitter_username
```

In my case, as I already had Bitlbee running on port 6667, I connected to port 6668:

```
/server localhost 6668 twitter_password twitter_username
```

Once you are connected to the tircd server, join the #twitter channel. tircd automatically imports everyone you are following, so they show up as users in the channel, and you also will see their recent posts. Any users that follow you back are voiced (+v).

Using IRC Commands with tircd

tircd works with a subset of traditional IRC commands, so it is pretty intuitive if you already are familiar with IRC. Your last Twitter status shows up as the topic of the #twitter channel, and if you want to update your status, all you have to do is type a message in the channel. If you want to send a direct message to other users, simply send them a private message. Likewise, if they send you a direct message, it shows up as a private message in IRC.

There are two different ways to follow or remove users. First, you can follow or remove them from the Twitter site or from any other Twitter client, and you will see those users join or leave your #twitter channel. Second, you can use the `/invite` IRC command followed by the users' Twitter user names to follow them. To remove them, all you have to do is `/kick` them from the channel. If you want to block users completely, just use `/ban`, and use `/unban` to unblock them. If you want to get information about a user, you can use the standard IRC `/whois` command.

Advanced Twitter Channels

One of the more interesting aspects of tircd is that you can set up multiple channels with only certain users in it. This can be useful if you follow a large number of accounts and want to organize them. Simply `/join` a new channel on the tircd server, and use `/invite` to add those particular users to that channel. Now, whenever those users update their status, it appears both in #twitter and in this new channel.

You also can use new channels for custom search queries. Again, `/join` a new channel of any name, and then use the `/topic` command to change the topic to the Twitter search query you want to use. All of the results of your search then appear in the channel.

Okay, I admit it, Twitter isn't so bad when you can access it inside IRC. I still think it's easier and faster to chat with people over IRC, but with tircd, I can find out what Larry King and Oprah had for lunch in my localhost #twitter channel and chat with all the great people in the official #linuxjournal channel all from the same client.■

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.

Ksplice Uptrack Service

Across the pond at Germany's big LinuxTag event, Ksplice unveiled Ksplice Uptrack, a new service that installs security and bug fixes on a running kernel without rebooting. Ksplice, whose technology was developed at MIT, claims to be the only solution that allows this application of updates without rebooting. Currently available for Ubuntu 9.04, Uptrack supports generic, virtual and server kernels. It also works in VMware, Xen, Virtuozzo or other virtualized environments. Although the initial release is a consumer-oriented solution, an enterprise solution is expected in Q3 2009. [See the August 2009 issue for a feature article on Ksplice.]

www.ksplice.com

Fixstars' CodecSys CE-10 H.264 Encoder

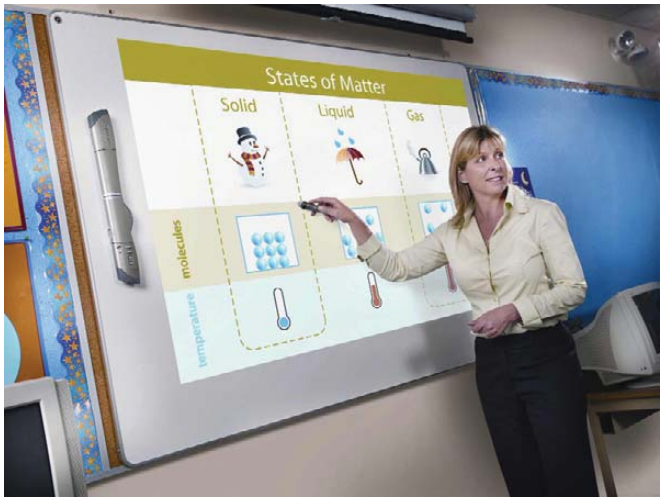
The company Fixstars (formerly Terra Soft of Yellow Dog Linux fame) keeps cranking out juicy goodies for the PS3 and other Cell-based platforms. The latest solution is the CodecSys CE-10 product, a faster-than-real-time H.264 video encoder that runs off a live USB stick or CD on the Sony PS3 using a microversion of Yellow Dog Linux as its live OS. In the solution, the PS3 acts as an external accelerator, encoding video files sent from the host PC, replacing expensive encoder cards and workstations. The PS3 is connected via Gigabit cable. The H.264 format allows for compression to half the size of MPEG2 for DVD and TV broadcasting while retaining equivalent quality.

us.fixstars.com/products/codecsys

Entuity's EYE NPE

Entuity's new network management solution, Eye of the Storm Network Professional Edition (or EYE NPE for short) is now available. The company says that EYE NPE offers "enterprise functionality at a price point previously reserved for workgroup-class tools" and adds "revolutionary technology for the mid-market" regarding automation, accuracy and deep functionality. Other product advantages include automatic surveying of networks in real time, an intelligent view of object connectivity, root-cause analysis and a broad range of configurable thresholds. Furthermore, rather than a device-centric approach, EYE NPE includes user-configurable views that represent any logical collection of devices or segmentation of the network. The solution runs on RHEL Server, VMware ESX Server and Microsoft Windows Server.

www.entuity.com



mimio Studio

Just in at the education desk is mimio Studio 6, the latest release of mimio's interactive teaching software. The application is one element—together with the mimio Interactive System, a projector and mimio Ink Capture Kit—in a system of presenting interactive lessons to students in schools. One can interact directly with the system, as well as import a wide variety of external media. New features in version 6 include support for multimedia files, a gallery for fast lesson creation and a slimmer toolbar. Interesting for us Linux geeks is that the system now works with Linux and Mac OS in addition to Windows, enabling a lesson created on one platform to run on all three.

www.mimio.com

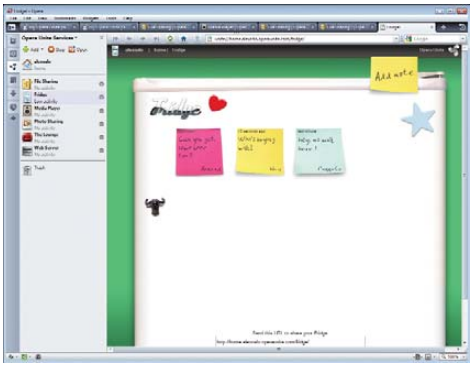
Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

CodeWeavers' CrossOver Linux, CrossOver Mac

Building on numerous improvements in the Wine Project, CodeWeavers has released version 8.0 of its popular CrossOver Linux and CrossOver Mac products. These products respectively transform Linux and Mac OS X into Windows-compatible operating systems for selected applications. Both products, says CodeWeavers, include support for Internet Explorer 7, Quicken 2009 and performance upgrades for Microsoft Office 2007, particularly Outlook. In addition, users will find that many other previously supported applications will run much faster and more stably. The company further offers that all of its products are significantly less expensive than the cost of a Windows license, allowing users economically and legally to eliminate the need for Microsoft Windows. (And there was much rejoicing.)

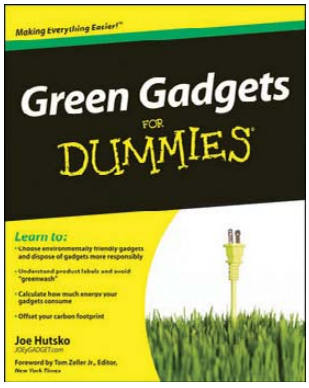
www.codeweavers.com

Opera Unite



The company Opera (of browser fame) says that its new Opera Unite is a new technology that will shake up the old client-server computing model of the Web and "decentralize and democratize the cloud". Essentially, Opera Unite is a Web server on the Web browser. It turns any computer into both a client and a server, allowing it to interact with and serve content to other computers directly across the Web, without the need for third-party servers. For consumers, Opera Unite offers greater control of private data and makes it easy to share data with any device equipped with a Web browser. For Web developers, Opera Unite services are based on open Web standards, and creating a full Web service is as easy as coding a Web page. Currently, Opera Unite offers six services: file sharing, Web server, media player, photo sharing, "The Lounge" chat service and "The Fridge" message exchange.

unite.opera.com



Green Gadgets for Dummies (Wiley)

Greening your gadgets and lifestyle can be not only fun but money-saving as well. Such is the motto of Joe Hutsko's new book *Green Gadgets for Dummies* from Wiley, a title billed as a friendly reference for exploring the environmental and financial benefits of green gadgets. Green gadgets encompass everything from iPods to energy-efficient home entertainment devices to solar laptop chargers and crank-powered gizmos. The book explains how to research green gadgets, calculate energy consumption, make a smart purchasing decision, use products you already own in a more environmentally friendly way, and bid farewell to electronics that zap both energy and money. Finally, the book covers product labels and how to avoid "greenwashing"—that is the overselling of environmental benefits.

www.wiley.com

CoroWare's Explorer

CoroWare Technologies announced the Explorer, an all-terrain robot designed and optimized for conducting R&D into new robotic applications that operate in unstructured, outdoor environments. Built on a ruggedized chassis, the Explorer functions well outside the lab, navigating rough terrain and resisting environmental elements. The Explorer's camera, wheel encoders and GPS enable the robot to examine the environment while the fully articulated four-wheel drive ensures the Explorer can navigate curbs, steps and inclines. By including a 2.0GHz PC-class processor, 80GB disk storage space and Ubuntu Linux with support for Player Project pre-installed, Explorer is ready to support any software the developer desires. Explorer comes standard with four-wheel drive, 802.11n Wi-Fi, GPS and 1600x1200 color camera. Expansion capabilities exist via extra USB, RS-232, digital I/O and analog inputs. Options include wheel encoders, a pan/tilt/zoom camera and a 64-bit dual-core motherboard.

www.coroware.com/explorer

Fresh from the Labs

Linux-MiniDisc—(Almost) Full MiniDisc Support for Linux

https://wiki.physik.fu-berlin.de/linux-minidisc/doku.php

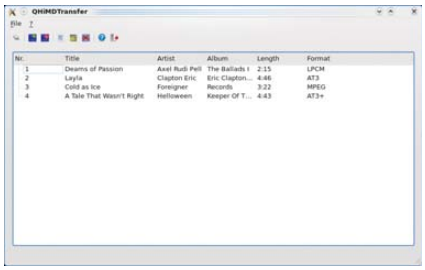
If you've got a stack of MiniDiscs lying around rotting, because you hate rebooting into Windows just to have basic access to *your* hardware, others exist who share your pain. One of these people is Adrian Glaubitz. Adrian sent me an e-mail, saying the following:

> Almost all newer MiniDisc-Walkmans made by Sony and other manufacturers have a USB-connector that allows download and upload of audio tracks and data to the MD-Walkman from a PC. However, since Sony is also a major record label, it has adopted a sophisticated system of DRM protection that requires a proprietary software from Sony called SonicStage, which runs on Windows only; even the latest Wine version is not able to run it smoothly enough to allow transfer from/to an MD-Walkman.

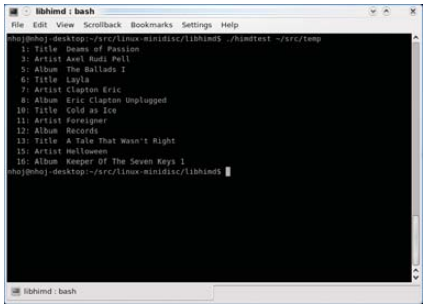
Being a passionate Linux user since 1998, Adrian was annoyed by always having to reboot into Windows to do anything with his player. There were Linux projects around, but they never allowed him to do much more than control his player—audio transfers were impossible. Adrian then decided to start this project, together with a friend who'd been working on parts of Wine for years (a great exercise in reverse engineering), and now the project has almost 20 people (some from older/defunct MiniDisc projects) contributing to the program in some form or another.

Installation First, there are many strange library requirements to take care of, so jump into your package manager to grab these elusive creatures (they might have different names in your distro, but the following at least should give you a clue):

```
libqt4-dev build-essential libglib2.0-dev
 ➥libmad0-dev libmcrypt-dev cmake libsox-dev
 ➥libmcrypt4 libmcrypt-dev cmake
```



The Linux MiniDisc Project now is at the point of basic GUI interaction and functionality with your player—a major milestone for previously neglected MiniDisc users.



Tech heads should have a look at some of the additional console tools available for further device wrangling.

To download the source, grab the repository using git. Open a terminal, and enter a directory where you won't mind the source being saved. Now, enter the following command:

```
$ git clone git://z6.physik.fu-berlin.de/linux-minidisc
```

This project is broken down into two major parts: libhimd (the library) and QHiMDTransfer (the GUI application). Let's compile both of them at once. Change into the linux-minidisc directory, and compile the program with the following commands:

```
$ cd linux-minidisc
$ cmake .
$ make
```

Take note of the . character after cmake; it's not a misprint, and you'll need it! Once compilation has finished, change into the QHiMDTransfer directory and run the program, like so:

```
$ cd QHiMDTransfer
$ ./QHiMDTransfer
```

Usage Once you're inside the application, you need to mount your MiniDisc device before you can browse it, upload to it or download from it. If you don't have a MiniDisc device, but you're still interested in exploring this program's features, there's an image you can use to simulate the device available on the given wiki page, along with instructions. When you have your device mounted, click File→Connect, and choose the folder under which your MiniDisc player is mounted.

If all goes well, your player's contents will come up in the main window. From here, you can choose to copy to or from the player with some fairly obvious cues from the GUI (it's a pretty basic interface). For those interested in doing more with their MiniDisc players, there are also tools like himdtest in the libhimd directory for things like track uploading, encryption info and so on.

For the moment, you can upload only MP3s and unencrypted PCM files as WAVs, but the team is working on total functionality. As Adrian told me:

> We are now very confident that soon we will have finished completely reverse-engineering the necessary protocols and file formats, so that there will be complete support for MiniDisc on Linux without any limitations by DRM, which are imposed by the original bloated Windows software. Once we have a first stable version, a friend of mine who is a Debian developer will help get the software into Debian and make it available to all Debian/Ubuntu-users.

I hope they do. The more niche hardware that's supported by Linux, the more our OS will be known for hardware-friendliness. Adrian tells me that he's also chasing some Qt programmers who can spruce up the GUI a little, so if you're a programmer on the lookout for a project to contribute to, give him a shout.

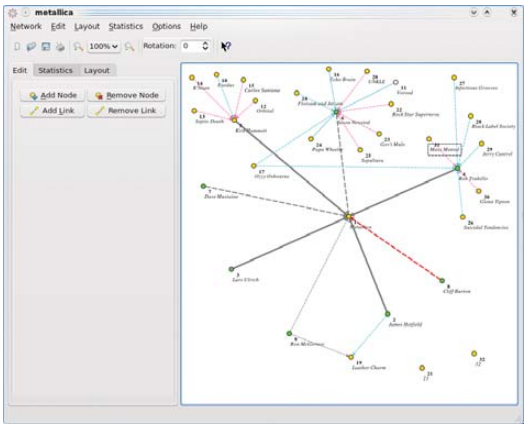
SocNetV—Social Networks Visualizer

socnetv.sourceforge.net

According to SocNetV's Web site:

> Social Networks Visualizer (SocNetV) is a flexible and user-friendly tool for the analysis and visualization of Social Networks. It lets you construct networks (mathematical graphs) with a few clicks on a virtual canvas or load networks of various formats (GraphViz, GraphML, Adjacency, Pajek, UCINET, etc.) and modify them to suit your needs.
>
> The application can compute basic network properties, such as density, diameter and distances (shortest path lengths), as well as more advanced structural statistics, such as node and network centralities (i.e., closeness, betweenness, graph), clustering coefficient, etc.

I tried reading that a few times and my brain exploded, so I thought I'd give it a look and find out just what it was all about and explain it in human language. What I discovered was a deceptively simple yet sophisticated program that organizes collected data in very cool ways. Now, I must state from the outset that it has nothing to do with social networking in the guise of MySpace, Facebook and so on (although you could use it for plotting those things out if you really wanted to). SocNetV is a means of plotting data in new and original ways.



SocNetV lets you plot data and shift it around, link sections and find constellations in the sea of information.

When you make your first few clicks, it appears to be just another basic plotting program, where you can make a flowchart or some other kind of information "tree". Not so. The advanced mathematical features turn grid points into a fluid, almost organic organism that can change and adapt in real time and reveal all sorts of patterns and flow in what appears at first to be stagnant information.

Installation If you head to the Web site's download section, SocNetV is available in packages for just about any distro you can shake a stick at, as well as a Windows binary, the usual source and even a Klik package (I haven't seen one of those for a while). I went with the Ubuntu package, but if your distro isn't on the list, or if you would prefer the source for whatever reason, you can do that too. If you are compiling from source, you need to grab the Qt4 development files, along with the QtWebKit development files. When you're ready, grab the source, extract it and open a terminal in the folder. From here, it's a case of doing the usual:

```
$ ./configure
$ make
$ sudo make install
```

Once the installation has finished, you can run the program by entering:

```
$ socnetv
```

If you're lucky, it'll also be in your system's menu; mine was under Education→Mathematics→Social Networks Analysis and Visualisation.

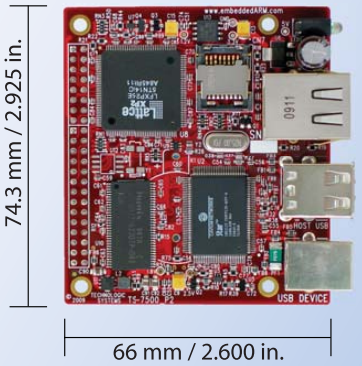
Usage Once inside, the first thing you'll see is a large blank white space, which is where your networks will be drawn. On the left are controls to Add or Remove a Node and to Add or Remove a Link. These are the most important controls, and you'll use them a lot. Now, let's create our first node.

Click Add Node, and a small yellow circle appears in the blank space on the right. This first node automatically becomes the first point of reference for all the other nodes, so it's best to make this node the most important—the nucleus, the genesis from

TS-7500
Embedded Computer

Faster. Smaller. Cheaper.

Qu. 100 $84



Powered by a
250 MHz ARM9 CPU

- Low power, fanless, < 2 watts
- 64MB DDR-RAM
- 4MB NOR Flash
- Micro-SD Card slot - SDHC
- USB 2.0 480Mbit/s host (2) slave (1)
- 10/100 Ethernet
- Boots Linux in less than 2 seconds
- Customizable FPGA - 5K LUT
- Power-over-Ethernet ready
- Optional battery backed RTC
- Watchdog Timer
- 8 TTL UART
- 33 DIO, SPI, I²C

Dev Kit provides out-of-box development + extra features

- Over 20 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products ship next day

Technologic SYSTEMS

We use our stuff.
visit our TS-7800 powered website at
www.embeddedARM.com
(480) 837-5200

which all the other nodes spring. With the node made, it's best to give this first node a label that sticks with the idea of it being a reference point.

Say you were mapping out your MySpace friends (goodness knows why, but let's run with it). You might want to name the first node something like "My Home Page". Or, let's say you were a Dr Who fan mapping out the Dr Who universe; you might want to call the first node "The Doctor", and so on. You can do this by right-clicking on the node, and choosing Options→Change Label.

Now, to add your surrounding nodes, click Add Node again, and a new node with the number 2 appears on the screen. To link this to node number 1, click Add Link. A series of prompts now appears in regard to the rest of the field of nodes, which is just the two for now. First up is the target node—1, by default. Next is the strength of the link, which, by default, will be 1.0. This value is very important, as it defines how valuable/important/relevant the link is to another node. You can use any number between –20 to 20, with positive numbers drawing a solid line between nodes and negative numbers drawing dotted lines. The higher the number, the thicker the line.

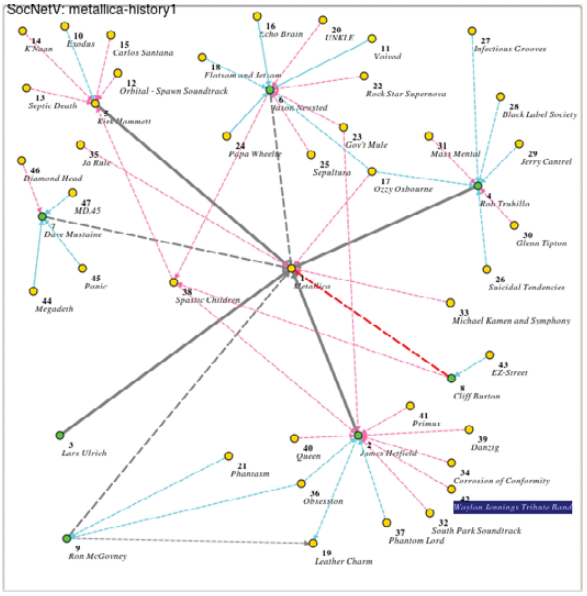
You've now connected your first two nodes, and from here I suggest adding some more to get the idea. If you right-click on a node, you'll notice the Options menu has a number of things to play with in terms of customizing each node, such as turning it into a square, changing the color and so on. Doing so helps differentiate one kind of node from another, helping to define what information it is representing visually. For instance, in my diagram of Metallica's history and affiliations (a band with a loaded history and a great deal of influence—a perfect testing ground for this kind of thing), band members are represented by a green circle, and bands/collaborators are represented by yellow circles.

You also can change the color of each line linking a node, adding more differentiation to a sea of probably messy information. For actual band members, I've gone with a strong gray line, with a dotted line for ex-members, and red line for the late Cliff Burton (RIP). Actual bands and important projects are signified by blue lines, and casual projects and one-offs are represented by pink lines. Don't forget that you also can move around nodes by left-clicking and dragging if things get messy and you need to do some rearranging.

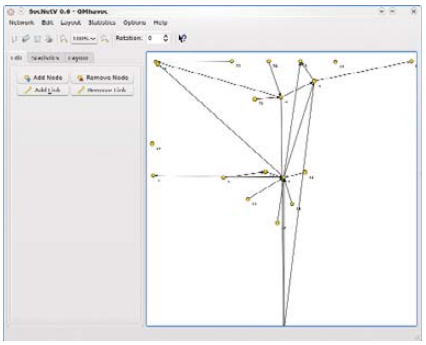
Once you've made yourself a full grid of information, you can apply a bunch of crazy mathematics that can morph your information in real time, showing you new patterns in the information that you probably never thought of before. Check out the Layout menu and experiment with all the options for a real demo, which showcases what this program is really all about.

Although this project still has a few kinks and interface problems, anyone interested in the flow of information and discovering patterns in any area of life definitely should check out this project. In terms of industry, social analysts looking for new patterns in society, wealth and so on would find this of particular use. I'd also like to try using it in Analytical Psychology, mapping out various constellations of ideas in someone's psyche. There are endless uses for a tool like this that are limited only by your imagination—fascinating stuff.■

What I discovered was a deceptively simple yet sophisticated program that organizes collected data in very cool ways.



My Metallica chart in all its glory! Well, there's a few mistakes, but I'm sure you won't mind.



Advanced mathematics can morph your networks around in real time (as this screenshot is in the middle of doing), as your structure begins to resemble a moving, almost breathing organism.

Certain preset formulas can constellate your information, showing you new information in otherwise banal data.

John Knight is a 25-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.

Brewing something fresh, innovative or mind-bending? Send e-mail to newprojects@linuxjournal.com.

ATLANTA LINUX FEST

DATE:

Saturday
September 19, 2009

WEBSITE:

atlantalinuxfest.org

COST:

Free!

ADDRESS:

IBM
4111 Northside Pkwy
Atlanta, GA 30327

All lovers of Linux and Open Source Software are invited to Atlanta Linux Fest: the place to learn, make new friends, and have fun!

Atlanta Linux Fest

- FOR MORE INFO LOG ON TO -

atlantalinuxfest.org

GOOGLE CHROME

the Making of a Cross-Platform Browser

Google's Evan Martin and Mads Ager discuss the challenges behind making a browser work well on Linux, Mac OS and Windows.

This article on the development of the Google Chrome cross-platform browser started off like any other interview. I interacted with Google by e-mail and phone and started pulling together the responses to my questions. It turned out that the "official" responses were much shorter than I was used to. "Why are these guys so shy?", I thought. In interviews, I typically have to whittle down my respondents' answers because they *love* telling their story—in glorious detail!

So, I went back to Google to see what was up. "Free your developers to speak!", I exclaimed. "We want to know the gritty insiders' take on Google Chrome development!" My contact there told me that interviews are challenging because a direct quote is like going "on record" and needs to be vetted by several layers of management (and maybe attorneys?). And, when you're the big fish in the pond, you have to be careful what you say. I am not used to such caution, and I certainly don't like it, but I indeed understand it.

After this and subsequent discussions, I realized we had a pretty complete picture of what Google Chrome is all about. The only hitch is that one part of the material came from direct interviews and another part came from more informal discussions and e-mail messages. Thus, we agreed that while I could talk freely about Google Chrome, only authorized material could be quoted. What follows, then, is a summary of my discussions with Google, followed by an interview with Google Chrome developers Evan Martin and Mads Ager. Martin is a Senior Google Software Engineer and Linux enthusiast working on all platforms of Google Chrome. Before working on Chrome, Martin worked on Google's search-result ranking. Mads Ager is Tech Lead for the V8 JavaScript engine project and and its integration in the Google Chrome browser; he is based in the Aarhus, Denmark, office.

JAMES GRAY

The Google Chrome Development Team
(Photo Credit: Jeromy Henry)

GOOGLE'S STRATEGY WITH CHROME

In some of my earliest conversations with Google, we talked about the company's motivations for building Chrome. After developing a range of rich and complex Web apps, the company saw that it was time to build a browser from scratch that could better handle "today's Web". From the beginning, they focused on a browser that innovated in four key areas, namely speed, simplicity, security and stability. Early on, the Google Chrome team realized that the linchpin for innovating in these key aspects, as well as to handling the new Web apps, would be a much more efficient handling of JavaScript. Thus, the V8 JavaScript engine, explained further below, was conceived and became central to the Google Chrome Project.

Google hopes that innovations like V8 will change the feel of the desktop, making the Web apps we're using more and more feel like native ones. Besides the internal code needed to achieve efficient JavaScript processing, Google Chrome hopes to maximize this native feel by keeping the UI minimal, including an "app mode" that lets one create a desktop icon that links to a URL with merely a tiny UI around the edges.

From a development standpoint, Google noted the difficulty in making this user experience acceptable on platforms with very different capabilities and conventions. Rather than just doing a brute-force port, the Google Chrome team has focused on often taking a step back from the code and looking at the larger picture of what a certain part of the code accomplishes for the user and then translated that into more abstract benefits for the respective Linux, Mac OS or Windows user. On some platforms, native capability exists in whole or in part for core functionality, such as sandboxed processes, but not on others. This fact has required a wide range of refactoring or writing new code depending on existing functionality found on the respective platform.

One example of making Google Chrome good on the Mac platform is what the company did with WebKit. The team first had to come to terms with what it meant to use WebKit for Chrome and determine what it could provide. Interestingly, Google says that in the examples of Chrome or Safari, only about half the code is WebKit. In addition, WebKit was never really designed to be run in a separate process from the rest of the browser UI. In order to accomplish this, Google had to write much of its own drawing and event handling "plumbing" rather than simply dropping a WebView into a window in Interface Builder. However, the developers have been able to draw on much of the work that was done for the Windows version to solve this problem.

Of course, Google Chrome's entire development process is much more efficient and potent given its open-source nature. More important than trying to "win the browser war" in the traditional sense—that is, get people to use Google Chrome as their primary browser—the company feels its open-source efforts with Chrome already have stimulated and seeded a great deal of innovation and made other browsers better than they would have been in Google Chrome's absence. In fact, Google takes at least some credit for speed improvements and security enhancements that have taken place in other browsers during the past year, which is advantageous for everyone.

Given that Google Chrome is open source, we were curious to know how involved outside developers have been to its development. Although my contacts were unable to give me specific numbers, I was told that outside participation is very high, especially in terms of bug reports from users of the early developer builds of the browser. Google also works very closely with the WebKit team, so changes made by WebKit developers at Apple or others in the WebKit community are integrated into Google Chrome as well.

And now, on to the interview with Evan Martin and Mads Ager.

THE DEVELOPERS SPEAK

JG: In a nutshell, what inspired Google to create Chrome and how did it come about?

EM: We built Google Chrome because we believed we could add real value for users and help drive innovation on the Web. Google Chrome is built for speed, has a very simple interface and uses innovative technology to ensure it is always secure and stable, providing a great experience for users as they browse the Web. But what's more, by making Google Chrome open source and developing a powerful new JavaScript engine, V8, we believe we can help spur innovation in the industry and provide developers with the platform with which to build the next generation of Web applications. This is good for users, and good for Google, as we benefit directly when the Web gets better.

LJ: What is the Google Chromium Project?

EM: After we wrote the code for Google Chrome, we open-sourced it under the name Chromium. Much like Firefox is a trademark of Mozilla, Google Chrome is a trademark of Google; the name Chromium is not, so distributions are free to use it to refer to the same project. We hope that developers and browser vendors take a look at the Chromium source code and that it will be useful for new projects built by the Open Source community in the future.

JG: This being our cross-platform development issue, we're curious to explore the challenges and innovations in that area. What have been the major issues in making Chrome great on all of its platforms?

EM: Much of the challenges we've encountered on Linux stem from how heterogeneous the user base is—which, surely, is also the strength of Linux. This ranges from how to port simple UI decisions (Chrome's shade of blue was chosen to look good next to the blue seen on every Windows computer), to getting boring technical details (a binary built on Ubuntu won't work on a Fedora machine), to real problems that will require engineering work to solve.

One good example of the latter is adapting our sandboxing model for Linux. Getting a process sandboxed in a way that's useful to us is challenging on Windows, with the relevant source code consisting of more than 100 files, but it needed to be implemented only once to work everywhere. On Linux, there are a variety of easier-to-use but different sandboxing systems available, and different Linux distributions ship

with different (or no) sandboxing APIs. Here's an article about a kernel patch we've proposed for discussion toward that end: **lwn.net/Articles/332974**.

JG: What innovations does Chrome bring to browsing?

EM: We did a lot of interesting things in building Google Chrome. First, it's simple and easy to use—we've designed Google Chrome to be as unobtrusive as possible, taking up the minimum amount of space on your screen, and allowing you to search and browse all from the address bar. Its multiprocessed architecture also ensures Google Chrome is fast and stable. Additionally, we designed Google Chrome for speed from the beginning, including building a new JavaScript engine called V8 from scratch to handle rich, complex Web applications.

JG: Can you tell us more about V8, its history, your rationale for developing it and who the key people were behind it?

MA: The V8 Project started in late 2006. At that time, existing JavaScript engines did not perform very well. The goal of the V8 Project was to push the performance of JavaScript engines by building a new JavaScript engine on which large object-oriented programs run fast. The V8 Project was pioneered by the dynamic duo of serial virtual machine builders Lars Bak and Kasper Lund in a farmhouse outside Aarhus, Denmark.

JG: What innovations and new approaches does V8 bring to the browser?

MA: V8 uses the concept of hidden classes and hidden class transitions combined with native code generation and a technique called inline caching to make property accesses and function calls fast. V8 uses precise generational garbage collection to make the engine scale to large object-oriented programs that use a lot of objects. In addition, V8 contains a JavaScript regular expression engine that was developed from scratch, is automata-based and generates native code for regular expressions.

JG: What language(s) is Chrome/V8 written in?

MA: V8 is mostly written in C++, but some of the basic JavaScript libraries are implemented in JavaScript itself.

JG: What platforms does V8 support?

MA: V8 runs on Windows, Linux and Mac.

JG: What CPU architectures does it support for native code compilation of JavaScript?

MA: V8 supports IA32 and ARM.

JG: Are there plans to extend it to other CPU architectures?

MA: We are working on a 64-bit version.

JG: Is the code generation better on some architectures than others?

MA: There are different trade-offs for the different architectures, and we try to make the code generators as good as we can for the different architectures. The code generator for IA32 does register allocation and does more inlining than the code generator on ARM. In general, the IA32 code generator has been tuned more than the ARM one.

JG: Did you name it after the internal combustion engine or the vegetable drink?

MA: The internal combustion engine. It was developed in the context of Google Chrome, and we thought that there should be a powerful V8 engine under the "chrome".

JG: Why did Google choose to develop a new JavaScript engine and use WebKit rather than use code from Mozilla?

EM: We have always been and remain great supporters of Firefox—Mozilla helped lead the way in much of the innovation we've seen in the browser space during the last couple years, with features like tabs, search boxes in the browser chrome and extensions. They've also proved that you can build a mass-market software product using open-source technology and collaborative development in the open. However, we initially thought of our work in this space as an experiment and didn't want to impose our ideas on anyone else. Rather, we thought about developing a new JavaScript engine and open-sourcing it so that other browser developers could benefit.

We think that numerous open-source projects are good for the entire space because they allow developers to make advances and share them quickly. We continue to have a great relationship with Mozilla, and many of our engineers actively work on features in Firefox through Mozilla's public participation process.

JG: What can you tell us about the status, road map and challenges regarding the Linux version? We're salivating here.

EM: The Developer version is available for a few Linux distributions already. Although this is an early release and not ready for your average user, we hope you get an idea of what Google Chrome for Linux will be like and keep following our development in the open as we make progress on a beta and stable version.

JG: How many developers in how many locations are dedicated to Chrome development, and how many solely to the Linux version?

EM: Although we don't go into details about the number of Google employees on any particular product, we have a core team of engineers who are working hard to get the Linux build of Google Chrome up and running. As a team, to prevent fragmentation, we try to have all developers work on all platforms—I refactor code on Windows to make it work on Linux, and if someone on the Mac team breaks the Linux build, it's his or her responsibility to fix it. Pieces like the networking stack can be worked on from any platform, so developers can just pick their preference.

At one point, I counted Google developers contributing from more than a dozen different locations (some work from their homes); we have even more once you count the contributions we receive from other developers. One of my favorite experiences of this project has been filing a bug one evening, then waking up the next day to see a patch to fix it from someone in Europe.

We've also received many patches from outside of Google, and have even promoted some of our best contributors to committers themselves.

JG: Was there a specific Google application that prompted Google to decide it needed a bigger/faster browser?
EM: I think of Google Chrome as being less about making Google applications faster and more about making the Web as a whole faster.

JG: What toolkits are used to build Chrome? And, are there any interesting issues regarding tools worth mentioning?
EM: Google Chrome on Linux relies on a ton of free software—about:credits lists more than 15 subprojects we include source from—as well as standard system libraries like FreeType, NSS (the Mozilla SSL/TLS implementation) and GTK+. There has been a lot of discussion on-line over toolkit choice; it was surprisingly uncontroversial within the team to choose the one that Firefox and Flash depend on and that we had more experience with. I think other options would have been just as good, and I would, in particular, love to see someone knowledgeable about Qt contribute patches.

Regarding tools, I'd like to especially call out gold, the fast linker that is little known but has been a lifesaver for us.

JG: How has the development of Google Chrome for Linux been going? Can you share some ups and downs you've experienced so far?
EM: I run only Linux at home. For me personally, the biggest up was after working on Windows for so long, to be able to install and use it finally myself.

JG: Is there a tentative date for when a beta release will be ready for Linux?
EM: Not yet, but we're working hard on it. You can track our progress on Linux development by running the in-progress version available at **dev.chromium.org/getting-involved/dev-channel** or via the mailing lists and source found on the Chromium developer site at **chromium.org**.

JG: Will the Linux and Mac OS versions one day catch up with and enjoy equal functionality with the Windows release?
EM: Yes, it is one of our highest priorities right now.

JG: Thanks to you both for your fascinating insights on Google Chrome!■

James Gray is *Linux Journal* Products Editor and a graduate student in environmental sciences and management at Michigan State University. A Linux enthusiast since the mid-1990s, he currently resides in Lansing, Michigan, with his wife and cats.

As we go to press, Google just announced its Chrome operating system. Chrome OS will be based mainly on Web applications and will add an intersting dimension to the "Google World", as it will be possible to run a completely Google-based desktop environment. Although the Chrome OS will be a separate OS, it will run Linux under the hood. We're not surprised. Keep reading *LJ* both here and on-line for more information on the Chrome OS and what it means for Linux users.

Donald L. Corbet "TheLinuxGuy®"
2009 Show Chairman

Keynote Speaker:
Mr. Robin "Roblimo" Miller

October 24, 2009
8 AM - 5 PM
Radssion Resort Orlando-Celebration
2900 Parkway Blvd
Kissimmee, Florida 34747

http://www.floridalinuxshow.com/

PLATINUM
O'REILLY OSCON Open Source Convention
GOLD
GSA PROPOSAL.COM
SILVER
O'REILLY®

Official Sponsors
BRONZE
SANGOMA
SHADOWSOFT
TSHIRTBORDELLO.COM Your place for dirty laundry

MEDIA
LINUX JOURNAL
LINUX PRO MAGAZINE
LinuxQuestions.org

Our Supporters

OSS Atlanta.org
Open Source Software

Florida Ubuntu Loco Team

Supporting Florida LUGs:
GatorLUG
Linux enthusiasts & professionals
Pensacola Linux User's Group
Florida Linux Users Group
Linux Users Group of Tallahassee
Greater Orlando LUG

Building the Linux Ecosystem

Listen to leading Linux technical minds build the innovations of tomorrow!

Participate in the development of the Linux ecosystem!

Amongst our invited speaker list:

- **Vivek Kundra, Federal CIO of the USA** (live by video)
- **Keith Packard, Intel, X Window guru** (confirmed)
- **Linus Torvalds, Advanced git tutorial** (confirmed)

The Linux Plumbers Conference brings key developers face-to-face to design and develop the Linux ecosystem – the kernel, core libraries, utilities and so on. We have an exciting roster of discussions and presentations – on hot topics under development as well as recent innovations. We encourage users and students to attend!

- Register today to get a lower price - fees increase September 1!

The Linux Plumbers Conference is organized by volunteers in assistance with the Linux Foundation and our corporate and media sponsors. For more information, visit us at:

www.linuxplumbersconf.org/

Rich Cross-Platform Desktop Applications Using Open-Source TITANIUM

The Titanium open-source platform lets Web developers leverage their Web skills for creating desktop applications.

MARK OBCENA

Titanium is an open-source platform that enables developers to build rich desktop applications using standard Web technologies. Titanium applications run natively on Linux, Mac OS X and Windows operating systems. At a high level, Titanium competes directly with Adobe AIR, although it differs from AIR in three major ways. First, Titanium is open source; it's licensed under the Apache Public License (version 2). Second, Titanium is fully extensible; Titanium extensions can be written using a number of popular languages, including C++, JavaScript, Ruby and Python. Finally, Titanium opens up user interface programming to popular languages like Ruby and Python—a job typically reserved only for JavaScript. Both Ruby and Python have full access to the Document Object Model (DOM), which puts these languages on par with JavaScript for building rich, dynamic user interfaces.

It is important to note that Titanium is not a system that provides a point-and-click ability to build a single application that runs both on the Web and on the desktop; however, that is not to say code sharing across the Web interface and desktop interface is impossible.

Some developers may choose to develop with a share-and-segregate pattern: write a common set of shared libraries, then write platform-specific code for use in a Web interface and other code for use in a desktop interface. In this case, you'll still have a single codebase, but you'll end up with two different apps.

Other developers may choose to develop using progressive enhancement. With progressive enhancement, you start by implementing a basic set of features, then as new resources become available, you build up functionality to make use of these new resources.

A good example is Google Docs. There's a basic set of features you can access on-line, but if you install Google Gears, you get off-line access and other features as well. The same goes for Titanium apps. Developers can enhance their Web applications progressively by adding features and functions that will be available only when the app is run on a Titanium instance. Using this approach you have just a single app.

Both of these techniques are valid choices when it comes to developing apps. Both techniques have pros and cons, and it's up to you as the developer to choose which method to use.

No matter which technique you choose—two separate codebases, one codebase and two apps, or one app—at the very least, Titanium allows you to leverage your Web development knowledge to build desktop applications. It lets you use HTML and JavaScript, as well as other languages most often associated with Web development, to develop desktop applications.

No More Limits on Web Development

Titanium is a development platform with one clear goal: leverage Web technologies to create rich, cross-platform desktop applications. Using Titanium, you can create desktop applications using HTML and JavaScript, yet still get features not available on browser applications. For example, Titanium Web applications built for the desktop can access the filesystem and interact with the underlying operating system.

The idea behind Titanium isn't new, but Titanium clearly separates itself by giving you something unique: unlimited possibilities with open-source choices. You aren't forced to use anything proprietary—you can use any library or framework you want. All technological decisions are yours to make.

Although I mainly program with JavaScript for Web applications, it isn't the only technology that powers the Web. Titanium works well with Python, PHP, Ruby, Java, Flash and Flex, and Silverlight. So whatever technology you're using right now to develop your Web applications, you'll likely be able to use it with Titanium.

Because Titanium is distributed under the open-source Apache Public License v2, you can download the source code, play with it, fork it and extend it. It's this extensibility that makes Titanium a platform that developers can grow with in the future. The platform can morph and evolve into different forms as new needs emerge.

Rapidly Evolving Web Development Platform

Titanium is evolving rapidly and has experienced several major changes to its architecture in the past few months.

The initial preview release of Titanium (PR1) incorporated WebKit and a modified version of Google Gears. Essentially, Titanium PR1 used WebKit as its main component, and additional features were exposed to the runtime via a native extensions system, which gave developers access to features from a modified version of Gears.

Soon after this initial preview release, the Titanium team started to re-architect the platform. Google Gears was removed, and instead, a new system for exposing new features was created: Kroll.

Kroll is the microkernel that powers Titanium and extends the framework. This compact microkernel, written in C++, is a cross-language, cross-platform "binding" and invocation framework that enables mixing and matching code within the kernel. All the features that Titanium exposes are accomplished via Kroll modules. By using Kroll, Titanium gains the ability to support a multitude of languages and technologies. And, because Kroll is fully extensible, anyone can add more features to the platform, using any technology. You don't need to be a C++ guru to extend Titanium. You can create new modules using Python and Ruby, or even just plain-old JavaScript.

Titanium's use of WebKit was retained during the rewrite from PR1, and for good reasons. Not only is WebKit one of the best, standards-compliant engines available today, but it also features lots of goodies, such as HTML5 client database storage, CSS transformations and animations, and a fast JavaScript engine. All of these, of course, are available on Titanium.

Getting Started with Titanium Applications

Enough theory—it's time to get our gloves out and start working with Titanium. First, download the latest version of Titanium from **titaniumapp.com/download**, and execute the package:

```
$ chmod +x Titanium-PR3-SDK.bin
$ ./Titanium-PR3-SDK.bin
```

You'll be greeted by a window containing the License Agreement; click I Accept to continue. Wait for the installer to finish, and the next thing that pops up on your screen is the second tool that you'll use the most while developing applications for Titanium (aside from your favorite text editor): the Titanium Developer.

Titanium Developer is the main tool you need for developing Titanium applications. It features several tabs for different purposes, ranging from packaging your app, managing projects and discovering cool stuff from other developers (Figure 1).

The first tab is Projects, which helps you manage your Titanium projects. Here you can create new projects, run them for testing and package them for distribution (Figure 2).

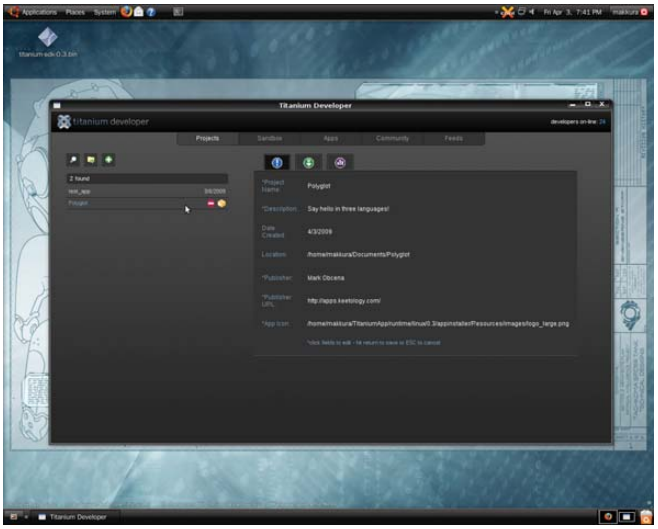


Figure 1. Titanium Developer

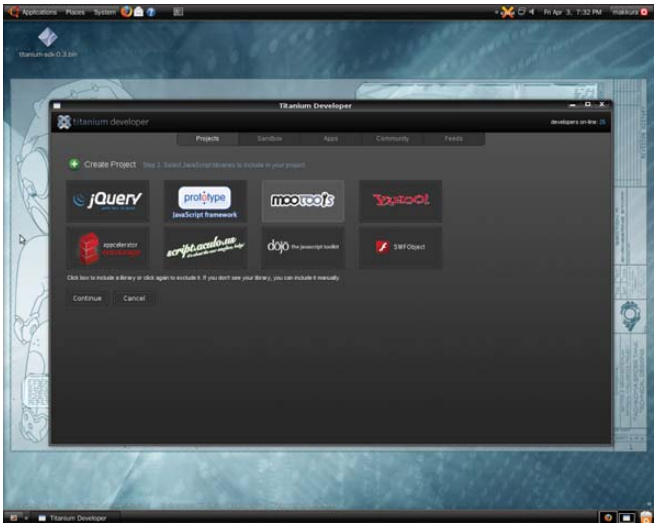


Figure 2. Projects

The Sandbox is a very useful tool for testing code without having to create a new project. To see it in action, try typing the following snippet of code into the text area on the Sandbox tab, then press Launch (Figure 3):

```
<div id="output"> </div>
<script>
document.getElementById('output').innerHTML
                          = "Hello World from Titanium";
</script>
```

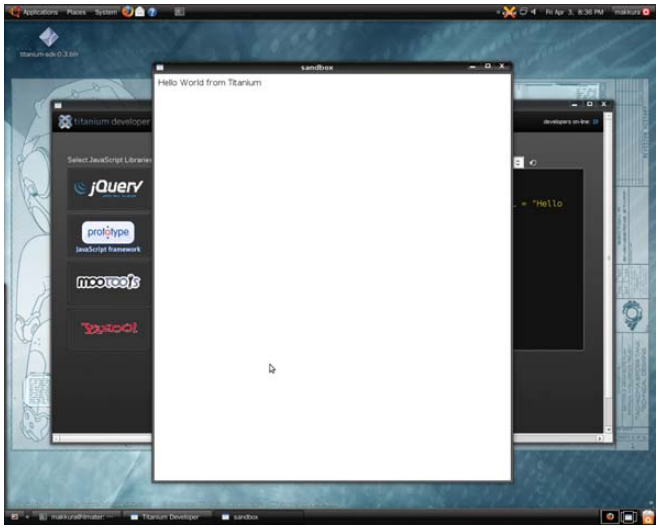


Figure 3. Sandbox

Congratulations! You just launched your first Titanium application via the Sandbox. Titanium Developer takes the snippet of code and launches a Titanium app using that code. On the side of the text area for the snippets, there are buttons for popular JavaScript frameworks. You can click one of those to add them to your sandbox application for testing instead of having to include them yourself.

The next tab, Apps, is a tool for discovering other awesome Titanium applications. Titanium enables you to distribute your applications easily via a distribution and packaging cloud. Every app that's packaged as public will be featured in the Apps tab, making it easier for you to share and distribute applications to your users (Figure 4).

Up to this point, Titanium Developer probably seems like any other development tool. You use it to create new projects, test them and package them for distribution—normal development stuff. But it goes beyond that. Titanium Developer isn't merely a development tool, it's a *social development* tool. Social development harnesses the power of social media for engaging, learning and collaborating with a global community of developers. That's why Titanium Developer contains two other tabs: Community and Feeds.

The Community tab contains a full IRC client that connects to the official Titanium IRC Channel, enabling you to talk to other developers, get help with tricky code or even share tips and tricks. The Feeds tab features live Web feeds from Twitter and FriendFeed regarding Titanium, so you'll be up to date about everything that's going on with the platform. The same

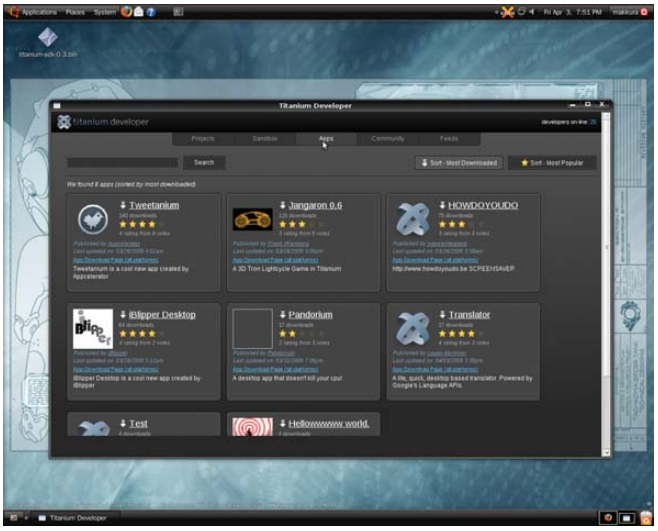


Figure 4. Apps

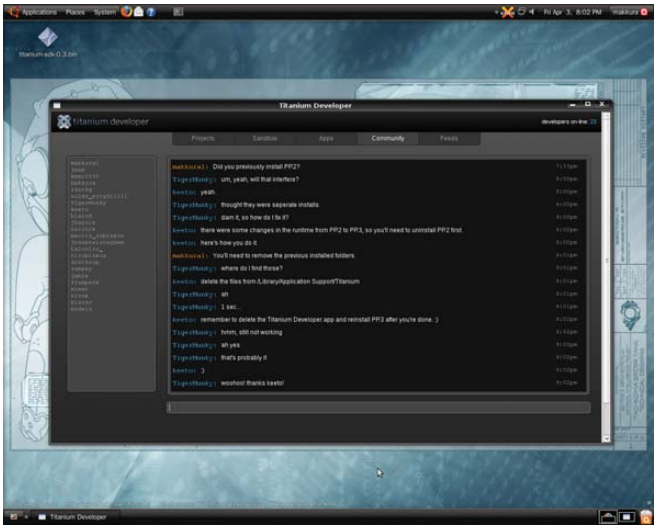


Figure 5. Community Feeds

tool that you use to create, package and deploy projects is also a tool that connects you to the worldwide community of developers working with the same platform as you (Figure 5).

And, the best thing about Titanium Developer? It's actually built using Titanium.

Titanium Project 101

It's easy to create new projects using Titanium Developer. The Projects tab takes you step by step into creating a new project. The first step asks you which JavaScript libraries you'd like to add to your project. You can add libraries by clicking on their logos, and you also can opt out by not clicking any. For this example, try adding MooTools.

Next, you need to input information about your project. Type anything you want for the Project Name, Project Description, Publisher Name and Publisher URL fields. For the Project Location, click on the small folder icon on the side of

the text field, and select the directory on your system where the project will be placed. Finally, you need an icon for the project, so click on the small folder icon on the side of the field for Application Image, and browse for an image file. Click Create Project.

The Project tab now displays your project on the list and some project information on a window on the right. This window has three tabs: the first one displays your project's information, the second one displays links to your application packages, and the third one displays distribution and download statistics for your application (Figure 6).

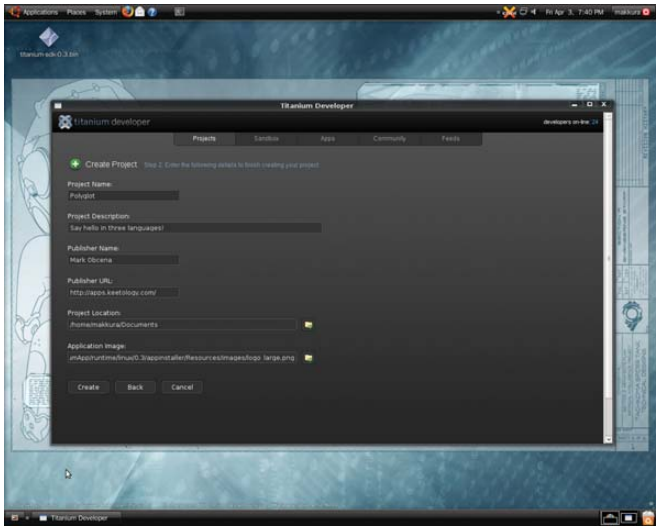


Figure 6. Create Project

Select the project in the list, and click on the small box icon. This brings you to the Packager where you can run, package or install your application. Click on the Package and Launch button to launch your newly created project.

You now have your first project, but it doesn't really do much at this point. So, open your file manager and navigate to the directory you selected for the Project Location when you created your project.

Inside your project directory, you'll see several files and directories (Figure 7):

1. dist: the directory where Titanium Developer stores your packaged app so you can launch it for testing.

2. manifest: a file used by Titanium to determine settings for packaging your apps, like your app's information as well as settings and versions of the modules you're using on your application.

3. tiapp.xml: the descriptor file for the application. This is used by Titanium to determine settings before running your application, such as settings for the initial window, version information and copyright information.

4. Resources directory: where all your application files are stored—your HTML pages, stylesheets and scripts all should be kept here.

Figure 7. Project Directory

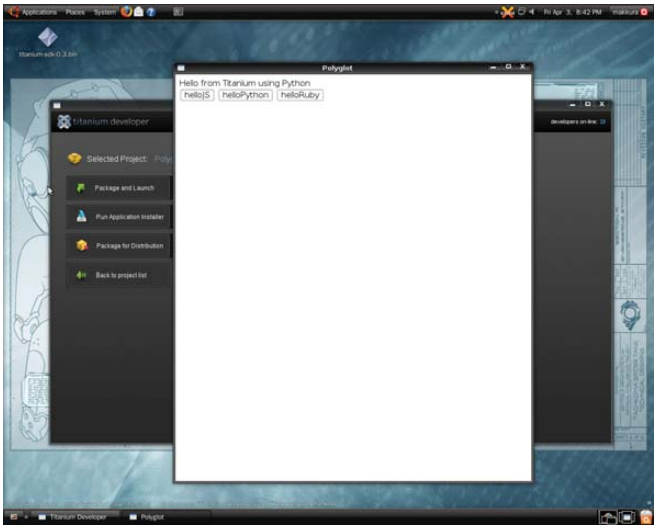


Figure 8. Hello World App

Titanium is very lenient when it comes to the structure of the Resources directory. You can create subdirectories to structure your project in any way you want, depending on your style.

Now, open the index.html file inside the Resources directory with your favorite editor. It should look like this:

```
<html>
<head>
<style>body{background-color:#292929;color:white}</style>
<script type="text/javascript" src="mootools-1.2.1.js"></script>
</head>
<body>
Welcome to Titanium
</body>
</html>
```

As you can see, it's just a simple HTML page, and Titanium Developer already included a link to the MooTools script

Listing 1. JavaScript, Python and Ruby, All in One HTML File

```
<html>
<head>
<script type="text/javascript">
function helloJS() {
    window.document.getElementById('output').innerHTML
                        = "Hello from Titanium using Javascript";
}
</script>

<script type="text/python">
def helloPython():
    window.document.getElementById('output').innerHTML \
                          = "Hello from Titanium using Python"

window.helloPython = helloPython
</script>

<script type="text/ruby">
def helloRuby
    window.document.getElementById('output').innerHTML \
                          = "Hello from Titanium using Ruby"
end

window.helloRuby = helloRuby
</script>

</head>
<body>
<div id="output">
</div>
<div>
<button onclick="helloJS()">helloJS</button>
<button onclick="helloPython()">helloPython</button>
<button onclick="helloRuby()">helloRuby</button>
</div>
</body>
</html>
```

(which also is included inside the Resources directory). Now, edit the file so it looks like Listing 1.

Save the file, then go to the Titanium Developer's Project tab, and click the package icon on your project. Click Package and Launch, and test your application. Click the buttons to get a hello world from three different languages—all in a single page (Figure 8).

While you're writing code, you're sure to run into bugs. Luckily, Titanium includes WebKit's Web Inspector, which you can use for various developments tasks. To open the Web Inspector, simply right-click on your app, and select Inspect Element.

Once you're done writing code and perfecting your application, you're now ready to package your application, which is easy to do with Titanium Developer. In the Packager window, click the Package for Distribution button.

You are given several options. The first one is to select for which platforms to package your app—you can choose from OS X, Windows and Linux (or all three). Next, you need to decide whether to bundle the runtime with your application or install it via the network during launch. Then, you decide which modules you'll add to your project and whether to bundle them with your app (Figure 9).

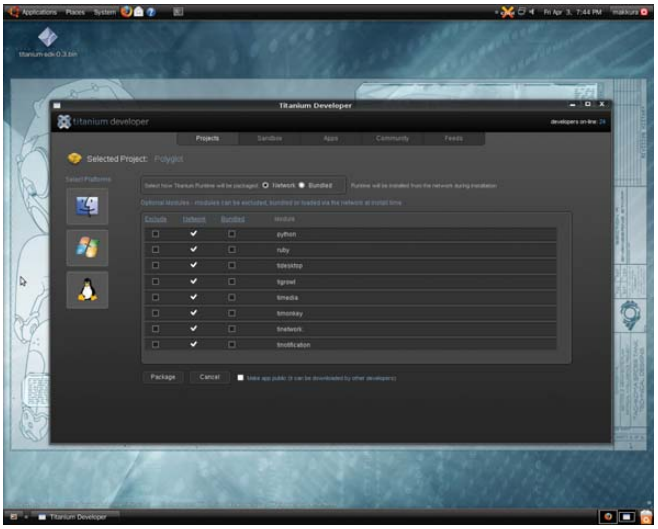


Figure 9. Packaging

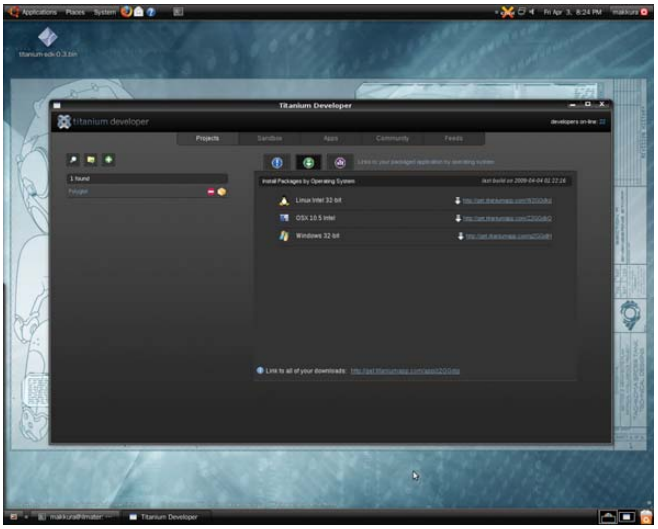


Figure 10. Stats and Links

Finally, you have the choice of making your project publicly available. By checking Make app public, your application will be added to the App directory and be made available to users everywhere. This helps immensely in distributing your application, because Titanium also hosts your files for you. When you're done, click Package.

Titanium Developer then uploads your project files to the Packager Cloud for packaging. When it's done, you are presented with links to your downloads for each platform you specified. If you made your app public, Titanium Developer also starts showing statistics for your application, such as the number of downloads for each platform and the user ratings for you application (Figure 10).

A Rich API for Rich Application Development

As you saw in the code above, all languages supported by Titanium have a window object. This is the shared global object and is used to bind methods and objects that need to be available on all languages. The main namespace for the Titanium API is also bound to this global object and can be accessed via window.Titanium.

Aside from WebKit goodies, such as client-side database storage and CSS animations, Titanium's current API also contains many of the necessary features needed for desktop application development:

- Titanium.Desktop: for launching third-party applications and opening URLs on the default browser.
- Titanium.Filesystem: for working with the filesystem for things like reading and writing files, creating and managing directories and so on.
- Titanium.Media: for working with media files, such as audio and video.
- Titanium.Network: for working with network-related tasks, such as socket connections and IRC clients.
- Titanium.Notification: for custom system notifications, as well as hooks to platform-dependent notification systems like Growl and Snarl.
- Titanium.Platform: for getting information about the user's system.
- Titanium.Process: for working with system processes, as well as launching and executing system commands.
- Titanium.UI: for working with native windows, menus and system chrome.

Unfortunately, going over all of these APIs would require an article (or two) in itself. Fortunately, the official Titanium site provides documentation with more details.

Getting Rich on the Desktop

Looking back, I wish that Titanium had already existed when my client asked me to do that project to store voice files. It would have saved me a lot of trouble fiddling with other solutions that couldn't truly satisfy the requirements of real, rich desktop development.

The good thing is that Titanium already is here, ready for action. You can download the SDK now, play with Titanium and join in on the community discussions to learn more about it.

Yes, Titanium might be a relatively new project. However, with the rapid rate of development I've seen so far, I'm planning to use Titanium to power the next generation of better, and certainly more powerful, desktop applications.■

Mark Obcena is a professional Code Sport player from Manila, Philippines. Aside from being a core contributor and platform evangelist for Appcelerator Titanium, he also contributes to several open-source, Web development–related projects. When he's not practicing his patented Backhand JS-Closure Attack, he writes about design, development and all things nifty for his site, Keetology (keetology.com).

SMALL, EFFICIENT COMPUTERS WITH PRE-INSTALLED UBUNTU.

GS-Lo8 Fanless Pico-ITX System
Ultra-Compact, Full-Featured Computer
Excellent for Industrial Applications

3677 Intel Core 2 Duo Mobile System
Range of Intel-Based Mainboards Available
Excellent for Mobile & Desktop Computing

DISCOVER THE ADVANTAGE OF MINI-ITX.
Selecting a complete, dedicated platform from us is simple: Pre-configured systems perfect for both business & desktop use, Linux development services, and a wealth of online resources.

ubuntu solution provider

LOGIC SUPPLY
www.logicsupply.com

LAZARUS for Cross-Platform Development

Lazarus may be the most native cross-platform development environment running on Linux, Windows and Mac OS X. Use it to create native applications with platform-independent code.

MATTIAS GAERTNER

Lazarus is an open-source library of visual components and a powerful IDE for rapid cross-platform development. The IDE contains all the features of a modern development suite, including a debugger, code completion, visual designers, refactoring tools, and translation and documentation tools. The Lazarus Project started on Linux ten years ago and now runs on all major platforms: Linux, Windows and Mac OS X. The Lazarus Project's motto is "Write once compile anywhere", and it provides cross-platform libraries, a cross-platform compiler and a cross-platform IDE.

Lazarus' features include the following:

- An easy-to-learn language: Pascal.
- A visual form designer.
- Producing native code executables that execute with speeds comparable to C/C++—no virtual machine here!
- Allowing direct access to system libraries.
- Supporting embedded assembler code.
- Easily handling big projects with millions of lines.
- Compatibility with the Delphi visual component library.

And, if all that weren't enough, Lazarus also is open source and free of charge, even for commercial development. The Lazarus IDE is shown in Figure 1.

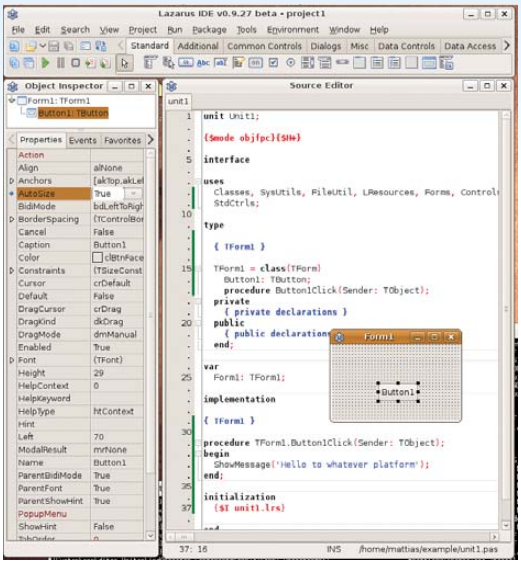


Figure 1. Lazarus IDE

Free Pascal Compiler

Lazarus uses the powerful Free Pascal Compiler, (FPC) which understands Object Pascal (a descendant of Pascal). Free Pascal (aka, FPK Pascal) is a 32- and 64-bit professional Object Pascal compiler. It is available for the following operating systems: Linux, FreeBSD, Mac OS X/Darwin, DOS, Win32, Win64, WinCE, OS/2, Netware (libc and classic) and MorphOS, and for different processors: Intel x86, AMD64/x86_64, PowerPC, PowerPC64, SPARC and ARM. You can find binaries, packages and daily snapshots at the Free Pascal and Lazarus Web sites (see Resources). Free Pascal creates native code executables, like C and C++, and uses the GNU tools and object format, so it can use C libraries directly, and, of course, C/C++ code can use FPC libraries. The speed and size of the created code is comparable to GCC.

FPC also compiles fast—normally more than 10,000 lines of code per second. That is because in Object Pascal, forward declarations are more limited than in C/C++. This saves a lot of time, even for small programs, and allows you to be more productive. After a while, you'll compile without thinking, just to highlight even obvious errors.

The Free Pascal Compiler itself is written entirely in Object Pascal. At the time of this writing, the compiler is at version 2.2.4.

Like its ancestor Pascal, Object Pascal is very easy to learn. C and Java programmers will understand most Pascal code without any tutorials. The language is very type-strict, and many code inconsistencies are spotted at compile time. This is especially useful for big projects, when a refactoring eventually is needed, and all affected places must be found. The compiler also warns when a statement works on the current platform but may fail on another—for example, when an expression works differently on 32- and 64-bit systems.

Lazarus Component Library and the IDE

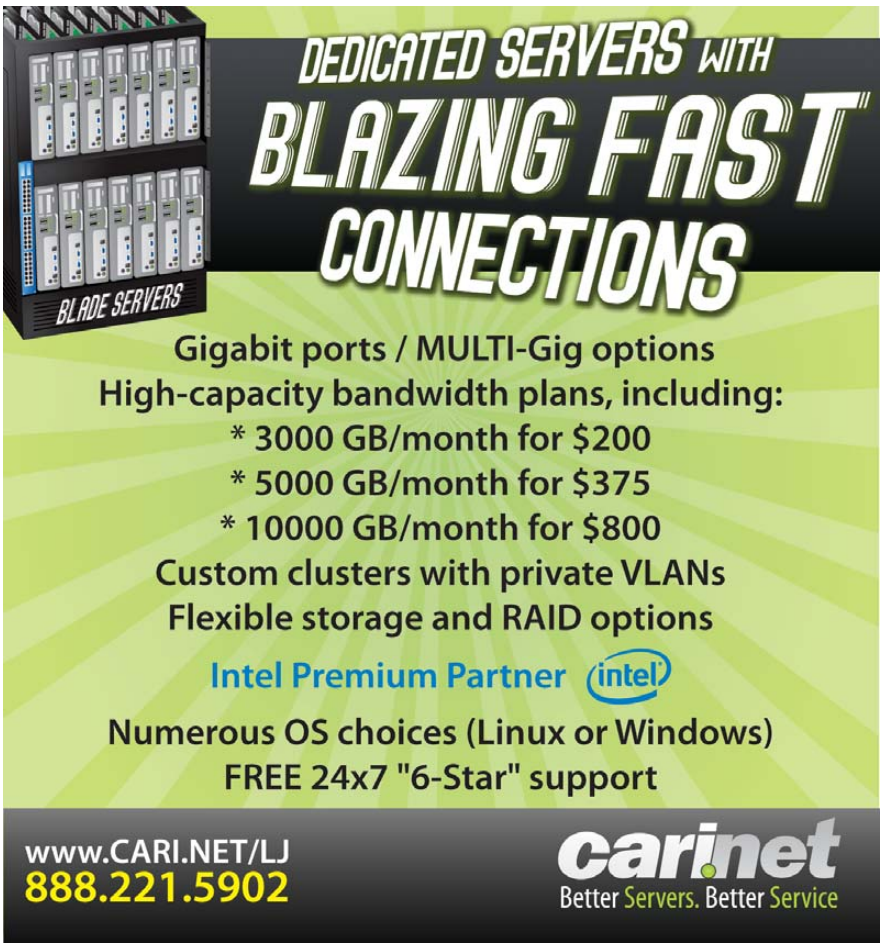
Lazarus gives FPC a face by providing the Lazarus Component Library (LCL), a library of visual components, such as buttons, edit fields, file dialogs and much more. These components run on Linux, MS Windows, Mac OS X, FreeBSD and Solaris using native widgets. Additionally, on Linux, you have the choice between GTK or Qt as a back end. The LCL calls the back-end widget sets and provides the glue between the platform-independent API and the widget set. The code itself needs to access only the LCL API, so no change is required when switching the widget set.

An LCL application compiled with GTK creates a native GTK application running on most Linux distributions out of the box. Under Windows, the choices are the WinAPI, GTK and Qt. For Windows CE, the back end is called wince. Under Mac OS X, the choices are Carbon, GTK and Qt. The widget set can be chosen automatically by the IDE or selected manually in the dialog for the compiler options. This allows you simply to copy a project developed under Linux to Windows and compile.

Some other LCL interfaces are under development—for example, fpgui, a widget set written completely in Object Pascal and Cocoa for the new Mac OS X libraries. So, if you don't care about native widgets and you want your application to look and feel exactly the same on all platforms, you can make use of the LCL and the fpgui library, which currently runs on MS Windows, MS Windows CE and Linux with X.

The Lazarus IDE uses the LCL and has an integrated visual form designer, which allows you to edit forms graphically, like Glade or Trolltech's Qt Designer. Lazarus' designer works directly with the corresponding Pascal unit source. For instance, double-clicking on a button in the designer automatically creates the OnClick in the source code and connects the button and the event handler. No further work is needed—

NOTE: FPC runs on more platforms than Lazarus. On those platforms, you can use the FP IDE, which runs in a terminal. The FP IDE usually is installed together with FPC and you can start it by typing `fp`.



simply compile and run. And, it works backward too. Remove a method from the code, and the IDE will disconnect it from the designed form.

The IDE even supports connecting two designed forms. That means a component on form1 can access the components on form2. No extra source code is required for this, just some mouse clicks.

The designer also allows you to inherit forms visually. For example, a base form can be created for all of an application's dialogs. Descendants can be created visually that inherit from this dialog. No extra source code is required. Even embedding a form into another form as a subcomponent can be done visually.

Of course, everything done in the designer can be done via source code at runtime too. The form data is stored in .lfm files, which are simple text files, so they are cross-platform also.

FPC: the Cross-Platform Foundation

Lazarus provides an outstanding native code solution. The compiler and most libraries are written with cross-platform in mind. That is why programs written in Free Pascal do not need to run a configure script before compilation. The base types, like char, byte, integer and string, work the same on all platforms. An integer always is a signed 32-bit value. The 64-bit integer is called int64. The native integer for a processor is called PtrInt for signed and PtrUInt for unsigned values. Lazarus itself can be compiled with a simple `make` or graphically in the IDE itself. And, of course, Lazarus is developed with Lazarus.

FPC's runtime library does not use libc; rather, it uses kernel functions, which change less often. Therefore, the created executables normally work on various Linux distributions and do not need to be recompiled for each new glibc version.

With Lazarus, you can write and debug the biggest part under Linux. But eventually, you'll need to test it on the other targets. However, you do not need to install Lazarus and all the development tools on all your target platforms. Cross compiling can be used to develop under Linux and target another operating system or processor. For example, you could develop under Linux and create Windows executables, and then test them with Wine or in a virtual machine running Windows, or on an actual Windows system. Cross compiling is a big time-saver, because it allows you to test on several platforms quickly and to use your favorite programs while developing.

Note, however, that cross compiling does require you to install the cross-compile tools and libraries, which can be tricky. Precompiled versions do not yet exist for all possible hosts and targets. Easy directions are provided for Linux to Windows, because of Wine, and for Windows to Windows CE, because there are installers with all needed tools.

Setting Up Cross Compiling

First, you need to cross compile and install the GNU binutils. This is well documented on several sites, including the Lazarus Wiki (see Resources). For many targets, this is as simple as downloading a single tar.gz and running a script with some parameters.

The next step is to cross compile the Free Pascal libraries. If you want to cross compile to another processor type, you need to cross compile the compiler too. Again, for many targets, complete scripts are available.

If your program requires third-party libraries, these must be cross compiled too. If they are written completely in Object

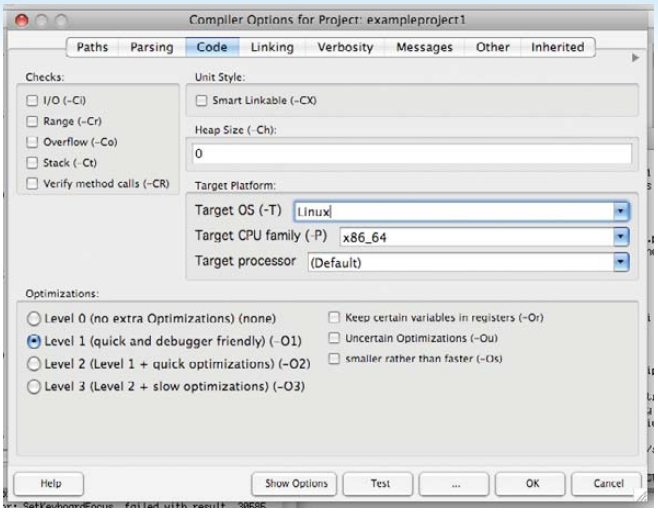


Figure 2. Lazarus Compiler Options

Pascal, normally you can just compile them. Lazarus will do that automatically for you. If they use system libraries, it can become difficult. The problems are then the same as for C/C++ compilers.

Once you've installed the cross compiler and libraries, cross compiling becomes easy in Lazarus. Simply pass the -T option to the compiler. For example, pass -Twin32 to compile a 32-bit Windows executable instead of a Linux binary. The -P option defines the target processor. Normally, you don't even need to pass special search paths, because of the path scheme used. For instance, the Pascal units for the fpc 2.3.1 compiler, for the processor type i386, and for target operating system Linux are installed under /usr/lib/fpc/2.3.1/units/i386-linux/. All filenames and search paths of the compiler and the IDE support macros, which greatly reduces the amount of command-line parameters and configuration settings.

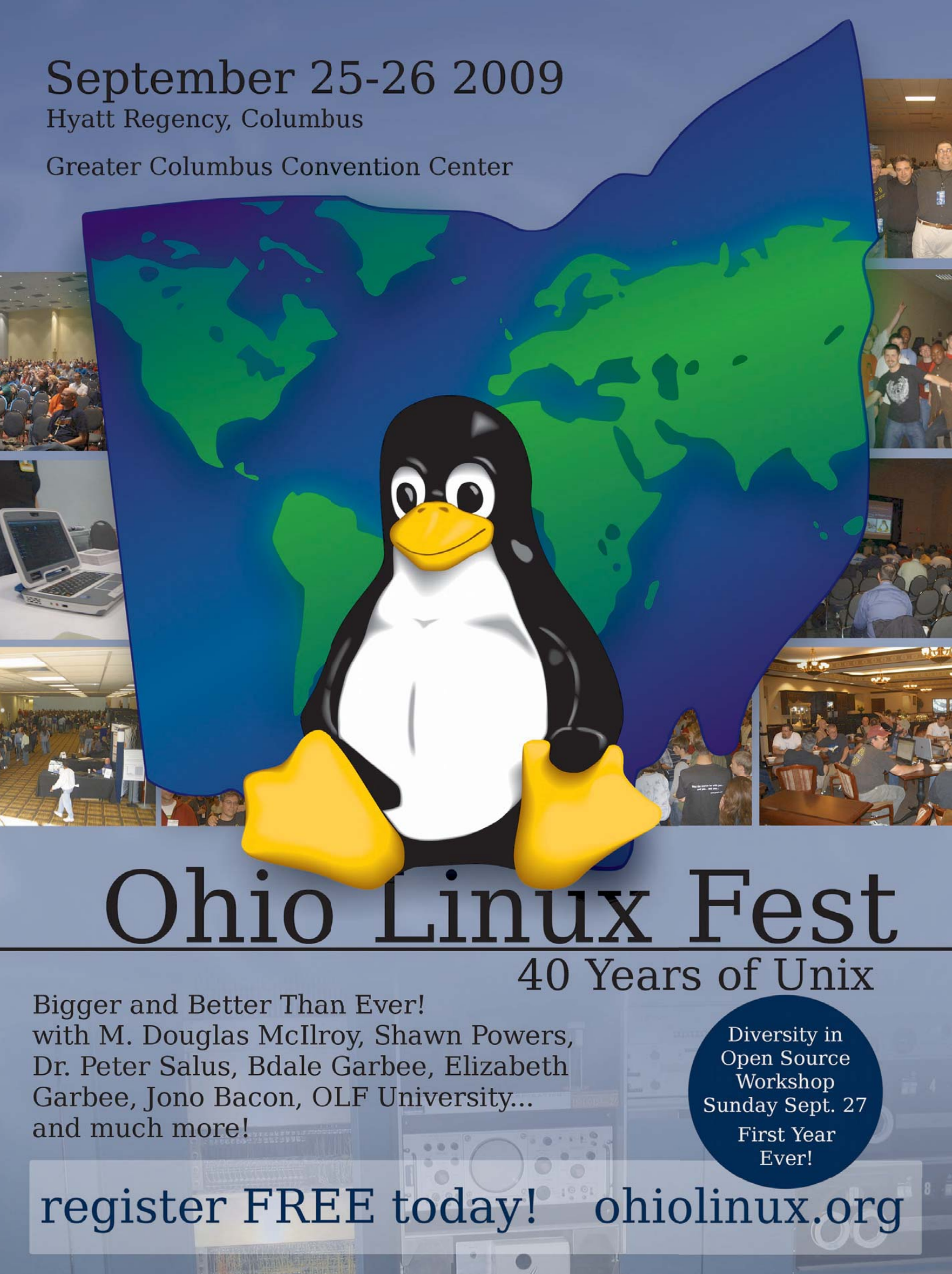
Lazarus reduces the amount of platform-specific settings even further. The IDE allows you to combine several source directories into a Lazarus package. A Lazarus package can be a library or just a logical module of a big project. A package has its own search paths, its own compiler settings and its own macros. All filenames and search paths are stored relative to the configuration file (.lpk file). A package can use other packages and inherit search paths and compiler settings. You can store a package anywhere on the disk. All search paths are adapted automatically on the fly. And, because every source has its own namespace, there is seldom a name conflict. You can switch to another version simply by opening the .lpk file. Each package also has its own output directories, normally one for each platform, which are created automatically.

When a package's source file is changed, the IDE automatically compiles the package and all packages in the current project that depend on it. You can fine-tune this automation for each package.

When you switch the target platform in the IDE, all packages' output directories are switched. The compiler options dialog is shown in Figure 2.

Code Completion and Cross-Editing

Most modern IDEs have some code-completion features. The IDE uses Codetools to parse the sources. Codetools is a library


September 25-26 2009
Hyatt Regency, Columbus
Greater Columbus Convention Center
Ohio Linux Fest
40 Years of Unix
Bigger and Better Than Ever!
with M. Douglas McIlroy, Shawn Powers, Dr. Peter Salus, Bdale Garbee, Elizabeth Garbee, Jono Bacon, OLF University... and much more!
Diversity in Open Source Workshop Sunday Sept. 27
First Year Ever!
register FREE today! ohiolinux.org

of parsers, search and refactoring tools and is independent of a specific compiler version. This allows the IDE to handle several versions of the compiler and to switch between them easily.

It also supports cross-editing. For example, it's possible to develop under Linux and write code for Windows. When a cross-platform compiler is installed, simply set the target OS to Windows in the IDE. The IDE code navigation and code-completion features now will work as if you were working under Windows. The following example illustrates this:

```
{$IFDEF Linux}
// write code for Linux here
{$ENDIF}
```

The curly brackets are compiler directives and work the same as C preprocessor directives. The code between the directives will be skipped by the compiler except when compiling for Linux. The IDE is a little bit smarter. When a different target operating system is active, the enclosed code will in most instances act like a comment. However, tools such as `find declaration` still will work within this code so that you don't have to switch the target too often. Some other macros that may be of use are:

```
{$IFDEF MSWindows}
// code for all kinds of windows
{$ENDIF}
{$IFDEF LCLGTK2}
// code when using GTK2 as widget set
{$ENDIF}
{$IFDEF big_endian}
// code for big endian processors like the powerpc
{$ENDIF}
{$IFDEF CPU64}
// code for 64 bit processors
{$ENDIF}
```

Generally though, high-level code doesn't need these macros, because the FPC system libraries provide most of the cross-platform functions and constants that you'll need. For example, the Lazarus IDE, with about 200,000 lines of code, uses them in less than 100 places.

Cross-Platform File Handling

There are numerous functions for cross-platform file handling that automatically use the correct path delimiter, case and other system specials. Instead of using the slash to separate directories, you should use the constant PathDelim. Under Linux, double path delimiters are treated as one, so you can concatenate filenames simply. This does not work on MS Windows, where empty directory names are allowed. Therefore, filenames should be normalized with one of the following functions:

- TrimFilename: removes leading and trailing spaces, combines double-path delimiters and does some minor cleanup.
- CleanAndExpandFilename: expands the ~ for the home directory under Linux, trims the file as above and chomps any trailing path delimiter.
- CleanAndExpandDirectory: works the same as CleanAndExpandFilename, but appends a delimiter if missing.

A very useful function is CompareFilenames, which compares two strings encoded in UTF-8 in the usual manner for the operating system. Under Linux, it compares them case-sensitively and distinguishes the various encodings of an a-umlaut. Under Mac OS X, the filesystem is usually case-insensitive, and all a-umlauts are normalized and treated the same. CompareFilenames does not check the actual filesystem, which might be case-insensitive. It is a quick compare function for sorting filenames.

When your application needs to store some configuration files, use the function GetAppConfigFile to get the standard directory. Under Linux, this is /home/username/.config/projectname/. For configuration files, standard formats like XML or INI files are recommended, which can be created by the easy-to-use classes TXMLConfig in the unit xmlcfg and TIniFile in the unit INI files.

Every operating system has its own idea of an application. Windows embeds the Explorer icon and version information in the binary. Since Windows XP, a manifest file can be added to enable theme support. Under Mac OS X, an application is

called an application bundle and is a directory with several XML and resource files. In the OS X Finder, the directory is shown as executable program, and the real files are hidden. A graphical application without this bundle file can be started but does not receive any input. The IDE automatically creates and updates these special files and structures for you.

Extending the IDE

Many packages extend the IDE with useful tools and graphical editors. Some examples follow.

For cross-platform OpenGL development, Lazarus provides a simple component named TOpenGLControl, which can be used on any LCL dialog. There are more-advanced third-party packages, like GLScene and Asmoday, that provide an object-oriented API for OpenGL.

You can write cross-platform dæmons that run under Linux as dæmons and under MS Windows as services with the lazdaemon package.

There are several cross-platform packages for databases. For example, the sqldblaz package provides cross-platform access and cross-database access to many common database systems.

Conclusion

This article gives a brief overview of how Lazarus and FPC make cross-platform development easy with a fast native compiler. Developers have the choice to optimize as far as they want, even down to assembly level or by accessing system libraries directly. The visual editors allow you to design dialogs and database applications quickly. The package system greatly simplifies the structuring of large projects and porting and distributing code to other platforms. The IDE cross-editing features allow developers to work under Linux and code for another target. ■

Mattias Gaertner joined the Lazarus Project in 2001, cutting his last ties to Windows and switching happily to Linux. Your comments are welcome at mattias@freepascal.org.

Resources

Free Pascal: **www.freepascal.org**

Lazarus: **www.lazarus.freepascal.org**

Multiplatform Programming Guide:
wiki.lazarus.freepascal.org/Multiplatform_Programming_Guide

Cross Compiling under Linux:
wiki.lazarus.freepascal.org/Cross_compiling_for_Win32_under_Linux

Cross Compiling with Lazarus:
wiki.lazarus.freepascal.org/Cross_compiling

How to Write Portable Code:
www.stack.nl/~marcov/porting.pdf

Advertiser Index

CHECK OUT OUR BUYER'S GUIDE ON-LINE.
Go to www.linuxjournal.com/buyersguide where you can learn more about our advertisers or link directly to their Web sites.

Thank you as always for supporting our advertisers by buying their products!

Advertiser	Page #	Advertiser	Page #
1&1 Internet, Inc. www.oneandone.com	1	Lullabot www.lullabot.com	21
Aberdeen, LLC www.aberdeeninc.com	33	Microway, Inc. www.microway.com	C4
Archie McPhee www.mcphee.com	78	Mikro Tik www.routerboard.com	3
ASA Computers, Inc. www.asacomputers.com	31	Ohio Linux Fest www.ohiolinux.org	63
Atlanta Linux Fest www.atlantalinuxfest.org	47	OpenSource World www.opensourceworld.com/live/12	77
Cari.net www.cari.net	61	Parallels, Inc. www.parallels.com	29, 79
Coraid, Inc. www.coraid.com	13, 79	Polywell Computers, Inc. www.polywell.com	71
Dell Computer Corporation www.dell.com/linux	5	Rackspace Managed Hosting www.rackspace.com	C3
Digi-Key Corporation www.digi-key.com	79	Saint Arnold Brewing Company www.saintarnold.com	78
Emac, Inc. www.emacinc.com	64, 79	SC World Congress www.scworldcongress.com	39
EmperorLinux www.emperorlinux.com	27	ServerBeach www.serverbeach.com	69
Florida Linux Show www.floridalinuxshow.com	52	Servers Direct www.serversdirect.com	9
Genstor Systems, Inc. www.genstor.com	37	Silicon Mechanics www.siliconmechanics.com	25, 35
iXsystems, Inc. www.ixsystems.com	7	StrayTats www.straytats.com	78
Linux Certified www.linuxcertified.com	41	Technologic Systems www.embeddedx86.com	45
Linux Foundation www.linuxfoundation.org	23	Ubiquiti Networks, Inc. www.ubnt.com	C2
Linux Plumbers Convention www.linuxplumbersconf.org	53	Utilikilts www.utilikilts.com	78
Logic Supply, Inc. www.logicsupply.com	59		

ATTENTION ADVERTISERS

December 2009 Issue #188 Deadlines
Space Close: September 28; **Material Close:** October 6

Theme: Embedded

BONUS DISTRIBUTIONS:
USENIX OSDI, RoboDevelopment, ISPCON Fall

Call **Joseph Krack** to reserve your space
+1-713-344-1956 ext. 118, e-mail joseph@linuxjournal.com

How to Be Cute on All Desktops with Qt

Qt always has been about cross-platform. By providing a rich API that isn't tied to a specific platform, Qt can be both intuitive to use and innovative.

The Qt toolkit originally was designed not only to be nice to work with, but also to allow for moving application source code between platforms. Today, the three major desktop environments are supported: X11, OS X and Windows. As portability is one of the key goals of the toolkit, it rarely runs into common issues, such as features missing on a specific platform or applications not integrating well in certain environments.

Qt's journey to fame really began more than ten years ago with the KDE Project. As one of KDE's cornerstones, it might come as a surprise to you that later incarnations of Qt try to integrate with GTK+ and GNOME. It even allows the incorporation of the glib event loop, all to fulfill the mission of providing portable code that looks and feels right on all platforms.

JOHAN THELIN

Looking Right

When discussing portable GUI source code, the graphical user interface is probably what comes to mind first. Providing widgets that look right on all platforms is an engineering feat. It takes many tricks to be able to use native painting methods, adapt to styling and just generally to fit in. Add to that the ability to subclass and customize widgets, and you have quite a handful of things that have to be incorporated.

And, making an application feel visually right on all platforms takes even more work. Margins, spacing, alignment—even the ordering of certain widgets—all need to be taken into account. Qt addresses all of these issues. A basic dialog window can be used to demonstrate how.

Figure 1 shows a property dialog with a set of labels to the left and fields for editing to the right. At the bottom are the standard Help, Apply, Ok and Cancel buttons. This might look like a simple dialog, but compare it to Figures 2 and 3. It's the same dialog, but on different platforms.

The platform imposes the order of the buttons at the bottom of the dialog, the alignment of the properties' labels, as well as the expansion policy of the fields representing the property values. All of these need to be handled according to the current platform's rules.

Providing widgets that look right on all platforms is an engineering feat.

Customizing the Look

In some situations, blindly following the current platform's look and feel isn't what you are after. Sometimes you may want to subtly give hints to the user. For instance, you may want to highlight all required fields or change the color of a progress bar. Usually, this means subclassing the source widget to specialize it. Then, you will use your special widget for all the required fields. Now, imagine having not only text fields, but also check boxes, drop-down lists and more.

In Qt, you can address this problem in two ways. Either you can create a custom palette object that you apply to all fields you want to highlight or change color. Or, you can use a stylesheet.

The advantage of using stylesheets is that they allow more advanced operations. Figure 4 shows this in three steps. The top row of widgets uses the standard style, and the second row uses the following stylesheet:

```
QLineEdit {
  background-color: rgb(255, 255, 185);
}

QCheckBox::indicator:unchecked {
  image: url(:/images/cb-unchecked.png);
}

QRadioButton::indicator:unchecked {
  image: url(:/images/rb-unchecked.png);
}
```

As you can see, the syntax was heavily inspired by the cascading stylesheets (CSS) used in Web design. The text field is an instance of the QLineEdit class. For it, it is enough to specify a background color. For the radio button and check box, you need to provide images that represent the indicator. More states than unchecked need to be included here, but to simplify for this example, they have been left out.

Merely changing the background color could have been achieved as easily by altering the specific widgets' palette. However, the last row in Figure 4 shows that you can go further. The stylesheet used here changes the font, text color, border and background. For the QLineEdit class, the stylesheet looks like this:

```
QLineEdit {
  color: red;
  font: 75 14pt "DejaVu Sans";
```

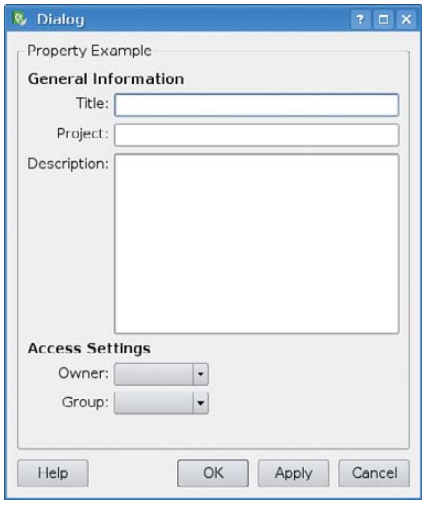


Figure 1. A Dialog in a KDE Environment

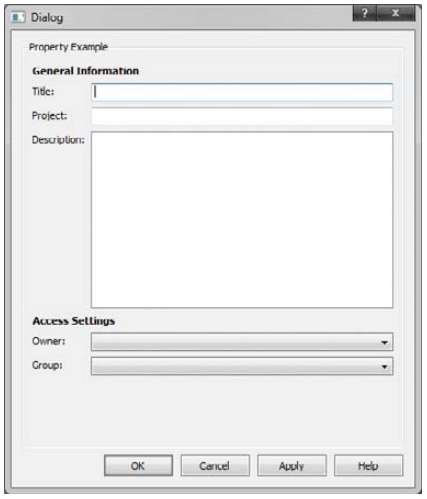


Figure 2. A Dialog in a Windows Environment

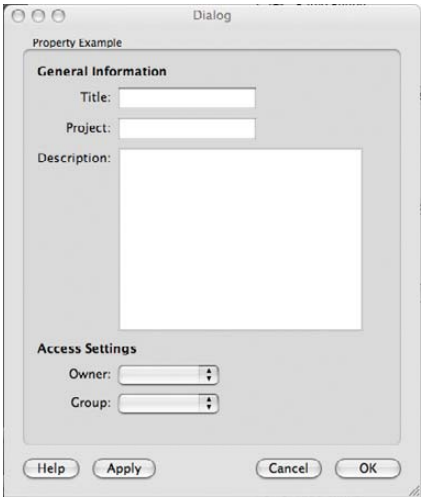


Figure 3. A Dialog in an OS X Environment

```
  border: 2px solid rgb(0, 112, 157);
  border-radius: 3px;

  background: qlineargradient(spread:pad,
              ➥x1:0, y1:0, x2:0, y2:1,
              ➥stop:0 black, stop:1 rgb(0, 112, 157));
}
```

As you can see, the color changes are not limited to only solid colors. The background is a gradient, and the whole shape of the border has been altered—all this, while still maintaining the source code's cross-platform portability.

Accessing Drives

What we've discussed so far affects only the visuals. You can try all of this from within Qt Designer or QtCreator without writing a single line of source code (not counting the stylesheets). But, cross-platform programming is more than just look and feel. For instance, how do you traverse a filesystem on multiple platforms without providing unique source code for each platform?

Qt provides classes for this. For example, the following short snippet shows the directories contained in the root directory of each drive of a given computer. On a Windows machine, it lists the drives one by one, while on UNIX machines, it lists only the root drive / (note that foreach is a Qt-supplied C++ macro):

```
foreach( QFileInfo drv, QDir::drives() ) {
  qDebug( "%s contains", qPrintable(drv.absolutePath()) );
  foreach( QString name,
           drv.absoluteDir().entryList( QDir::Dirs ) ) {
    qDebug( " %s", qPrintable(name) );
  }
}
```

By using the QDir class to access the filesystem, you can do so in a platform-independent manner. The class contains static functions for common entry points, such as drives, the user's home directory, the current directory, as well as the system's directory for temporary files.

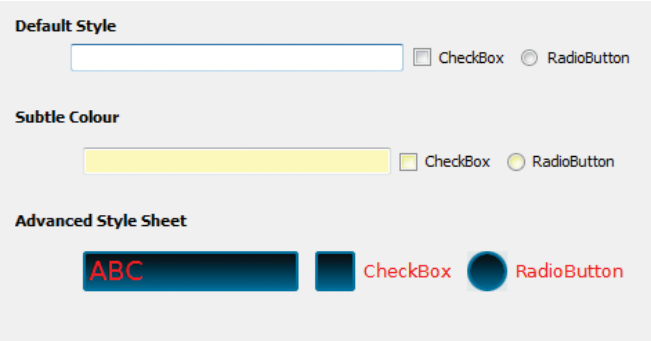


Figure 4. From Standard Style to the Extreme

Another common source of cross-platform problems occurs at a much more basic level—the encoding of text and data.

Handling Text

Another common source of cross-platform problems occurs at a much more basic level—the encoding of text and data. Qt provides a custom class for handling text strings called QString. It provides Unicode representation across all platforms. The string class itself can convert to and from UTF-8, ASCII and Latin1. It also can convert to and from most other string representations using text codecs. Qt comes with a variety of

Using Platform-Specifics through a Movable API

Qt might provide a cross-platform API that can cover almost all cases, but you still might want to use platform-specific features. For instance, opening the window as maximized in Windows and normal on OS X and X11. To handle these situations, Qt provides preprocessor defines describing on which OS you are running and which windowing system you are using. For example, on Linux, you'll find Q_OS_LINUX and probably Q_WS_X11.

When you know on which system you are running, you can access all X11 events by re-implementing the x11EventFilter function of the QApplication class. On OS X, you can get hold of the CoreGraphics handle from the macCGHandle function of each QWidget.

If you want to avoid writing platform-specific code, you still can give platform-specific hints. For instance, you can give a hint to a QDialog that it is a sheet. This is a dialog that appears inside another window or dialog that provides part of the larger window's features. You do this by setting the window flags of your dialog to Qt::Sheet.

On X11, this type of hint relies on the window manager's ability to understand it. This means the hint must be used as a hint, not a setting. If you want complete control, pass Qt::X11BypassWindowManagerHint. This tries to avoid the window manager completely, which is not a nice thing to do, but might be necessary.

codecs, but it also is possible to create custom codecs to handle special cases.

When reading and writing text to and from files, the encoding is respected by using the QTextStream class. This class provides a stream interface based on the << and >> operators. It usually autodetects the encoding, but you can use the setCodec function to force it to a specific setting. To illustrate, the following short snippet of code reads a line from a text file encoded as UTF-32 on a big-endian system:

```
QTextStream stream( &file );
stream.setCodec( QTextCodec::codecForName("UTF-32BE") );
QString myString = stream.readLine();
```

Which End Is First?

Speaking of endianness, this is often an issue that occurs when dealing with cross-platform code. The issue with endianness is that when you write binary data, such as a 32-bit value (four bytes), you can choose to write the bytes in two different directions: left to right or right to left, aka big endian and little endian.

The default order for writing bytes depends on the endianness of the system on which the program is running. Some architectures, such as IA32 and the VAX, use little endian. Others, such as PowerPC, ColdFire and SPARC, use big endian. Others still, such as ARM, MIPS, IA64 and Sparc V9, are able to do either (although which one is used often has to be hard-wired into the system when the hardware is built). Systems based on most of these architectures are commonly targeted by Qt.

To ensure cross-platform compatibility for binary data, you need to specify the order explicitly when writing and again when reading. By using a QDataStream to handle binary file formats, endianness no longer is an issue. You simply specify the byte order to use and then use the stream operators, and it just works.

The snippet of code below shows this. It also contains the setVersion function, letting you specify which version of Qt's encoding of complex data types you want to use. For instance, if the internal representation of colors changed between version 2 and 4 of Qt, by specifying an older version, you still can read and write data in the old format using the same stream class. This is something that comes in handy when having to handle old legacy file formats from modern code:

```
QDataStream stream( &file );
stream.setByteOrder( QDataStream::BigEndian );
stream.setVersion( QDataStream::Qt_4_0 );
int value;
stream >> value;
```

Storing Preferences

When handing user preferences, Windows has the registry. UNIX systems usually rely on hidden directories, one for each application. OS X has an XML format for preferences. This works fine for users. They usually do not rely on being

able to move their preferences between their computers, especially if they do not use the same operating system. From a software developer's perspective, the situation is different.

To resolve this, Qt provides the QSettings class. It provides access to each platform's preferred method. It also can be used to create and read INI files outside the platform's system that can be moved between platforms by the users.

The QSettings class relies on the name of the application and the application provider. Then, you simply use the setValue and value functions to write and read. The returned value is of the type QVariant. This type can be used to hold any type of data. The basic types, such as integers, are handled directly. More complex types, such as QColor, rely on the data stream operators:

```
QSettings settings( "The App Company", "The App" );
int v = settings.value("myInt").toInt();
QColor c = settings.value("myColour").value<QColor>();
```

Many more issues arise when moving code between platforms. Qt's solution is to provide a Qt API. This API removes almost all traces of specific platforms, while trying to support all functionality on each of the platforms involved. More complex cases than the ones shown here involve multithreading, database access, networking and so on.

Embedded

So far, this discussion has focused only on moving code between different desktops, which is just half of Qt's ambition. Qt comes in three embedded flavors: embedded Linux, Windows CE and Symbian S60.

The Windows CE and S60 ports make it possible to run Qt applications on phones and palmtops. Each of the ports takes the target device's styling into account and integrates the application in a seamless manner. At the time of this writing, the S60 port is available only as a technical preview; a full release is planned later in 2009.

The embedded Linux version makes it possible to run Qt directly on the framebuffer. This greatly reduces the footprint of the system, making it embeddable. The windowing needs are covered by an integrated window manager QWS (Qt Windowing System), but generally, these systems run their applications in full-screen mode.

One interesting feature is the ability to run applications in a virtual framebuffer, making it possible to emulate the correct resolution, bit depth and input behavior on a development machine. This allows you to start developing the software earlier in the project cycle. It also can simplify debugging, as you can avoid remote debugging.

The step when moving from desktop to embedded is generally larger than when moving between desktops or embedded systems. There are a number of issues that a framework cannot solve. The most common issues are available screen space, lack of computing power and lack of memory. All these areas are becoming less of a concern as the power, memory and screen resolution of embedded systems increase.

Qt provides the ability to style and stretch interfaces to fit the screen. You also can set the global strut. This is the minimum size that any user interface element can have. By adjusting this factor, you can tune widgets to make them usable using a finger, stylus or mouse.

Embracing Qt

Qt provides an API that can be used across a variety of platforms. All major desktops are supported, but also the major embeddable platforms. The strength of Qt is that all these platforms can be reached through one API. The API is provided by one library, one set of goals and one approach to constructing APIs. To take full advantage of Qt's cross-platform ability, you should embrace the use of Qt in all fields. If you do, you can move your code as easy as you can compile it.■

Johan Thelin has worked with software development since 1995 and Qt since 2000. Having seen server-side enterprise software, desktop applications and Web solutions, he now works as a consultant focusing on embedded systems. Johan can be contacted at johan@thelins.se.

Cross-Platform Development Using a Cross-Platform Environment

Qt comes with a set of tools that can be used separately or from within the fairly new QtCreator application. QtCreator was created using Qt and provides an advanced code editor, documentation, an integrated version of Qt Designer and editors for Qt-specific files, such as project files and resource files.

Because all the Qt tools also are available separately, it is common to use another IDE or just a text editor and command line. Qt Software provides integrations for Microsoft Visual Studio, Xcode and Eclipse. There also are a range of free IDE projects out there, such as Edyuk, QDevelop and KDevelop.

So, what does QtCreator provide that the others don't? First, it comes as a part of the Qt SDK. The SDK version of Qt comes as a single download with a prebuilt version of Qt and QtCreator set up and ready to go. Second, it provides a graphical debugger interface, letting you use gdb in the easiest possible manner across all desktop platforms supported by Qt. The debugger knows of Qt and provides macros for easy viewing of QString objects as well as for looking inside Qt's list classes.

Polywell Linux Solutions

More Choices, Excellent Service, Great Value!

Serving the Industry for More Than 20 Years

Quiet Performance NAS Storage

8TB $1,999

12TB $2,999

24TB $5,999

- Dual Gigabit LAN
- RAID-5, 6, 0, 1, 10
- Hot Swap, Hot Spare
- Linux, Windows, Mac
- E-mail Notification
- Tower or Rackmount

NVIDIA GeForce

Silent Eco Green PC

NVIDIA® nForce Chipset and GeForce Graphics
Energy efficient, Quiet and Low Voltage Platform. starts at **$199**

The Best Terminal PC

LD-001

Fanless Silent Mini-ITX PC

1G DDR2, Solid State Drive starts at **$299**
Nvidia Ion Based PC is now Available: ITX-9400

ITX-10A

ITX-30A with PCI Riser

ITX-20A with Slim DVD

ITX-1000 with WiFi Option

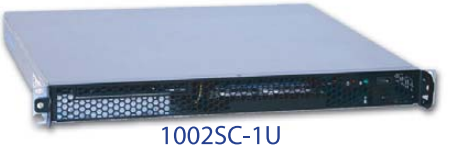
4U-24Bay 48TB Storage Server

Hardware RAID-6, NAS/iSCSI/SAN Storage
Mix SAS and SATA, 4 x GigaLAN or 10Gbit LAN

4024SS

Mini-1U Server for Data Center ISP

Intel Dual-Core or Quad-Core Processor, Dual GigaLAN
4GB to 8GB RAM, 2 x 500GB RAID HD
Linux Server Starts at **$499**



1002SC-1U

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

NVIDIA, nForce, GeForce and combinations thereof are trademarks of NVIDIA Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

POLYWELL

Open-Source Compliance

A discussion of open-source compliance, the challenges faced when establishing a compliance program, an overview of best practices and recommendations on how to deal with compliance inquiries. IBRAHIM HADDAD

Traditionally, platforms and software stacks were built using proprietary software and consisted of various software building blocks that came from different companies with negotiated licensing terms. The business environment was predictable, and potential risks were mitigated through license and contract negotiations with the software vendors. In time, companies started to incorporate open-source software in their platforms for the different advantages it offers (technical merit, time to market, access to source code, customization and so on). With the introduction of open-source software to what once were purely proprietary software stacks, the business environment diverged from familiar territory and corporate comfort zones (Figure 1). Open-source software licenses are not negotiated agreements. No contracts are signed with software providers (that is, open-source developers). Companies now must deal with dozens of different licenses and hundreds or even thousands of licensors and contributors. As a result, the risks that used to be managed through license negotiations now must be managed through compliance and engineering practices.

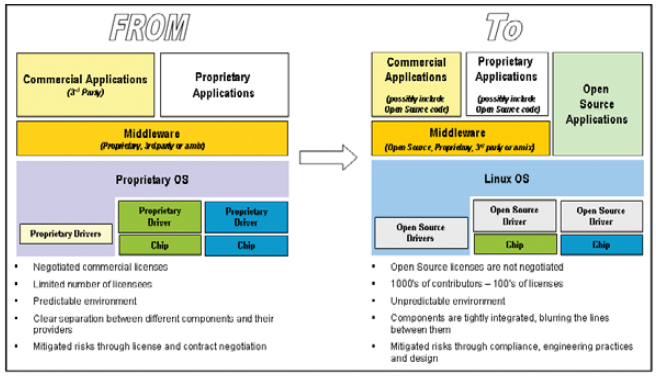


Figure 1. A new computing environment necessitates open-source compliance due diligence.

Enter Open-Source Compliance

Open-source software initiatives provide companies with a vehicle to accelerate innovation through collaboration with a global community of open-source developers. However, accompanying the benefits of teaming with the Open Source community are very important responsibilities. Companies must ensure compliance with applicable open-source license obligations. Open-source compliance means that open-source software users must observe all copyright notices and satisfy all license obligations for the open-source software they use. In addition, companies using open-source software in commercial products, while complying with the terms of open-source licenses, want to protect their intellectual property and that of third-party suppliers from unintended disclosure.

Open-source compliance involves establishing a clean baseline for the software stack or platform code and then maintaining that clean baseline as features and functionalities are added.

Failure to comply with open-source license obligations can result in the following:

- Companies paying possibly large sums of money for breach of open-source licenses.
- Companies being forced by third parties to block product shipment and do product recalls.
- Companies being mandated by courts to establish a more rigorous open-source compliance program and appoint an "Open-Source Compliance Officer" to monitor and ensure compliance with open-source licenses.
- Companies losing their product differentiation and intellectual property rights protection when required to release source code (and perceived trade secrets) to the Open Source community and effectively license it to competitors royalty-free.
- Companies suffering negative press and unwanted public scrutiny as well as damaged relationships with customers, suppliers and the Open Source community.

FSF Compliance Lab

The Compliance Lab at the Free Software Foundation (FSF) helps enforce the license for all free software. Information about the life cycle of compliance cases handled by the FSF is available at **www.fsf.org/licensing/compliance**.

Lessons Learned

There are three main lessons to learn from the open-source compliance infringement cases that have been made public to date:

1. Ensure that your company has an open-source management infrastructure in place. Open-source compliance is not just a legal exercise or merely checking a box. All facets of a company typically are involved in ensuring proper compliance and contributing to the end-to-end management of open-source software.

2. Make open-source compliance a priority before a product

ships. Companies must establish and maintain consistent open-source compliance policies and procedures and ensure that open-source license(s) and proprietary license(s) amicably coexist well before shipment.

3. Create and maintain a good relationship with the Open Source community. The community provides source code, technical support, testing, documentation and so on. Respecting the licenses of the open-source components you use is the minimum you can do in return.

Compliance Challenges

Companies face several challenges as they start creating the compliance infrastructure needed to manage their open-source software consumption. The most common challenges include:

1. Achieving the right balance between processes and meeting product shipment deadlines. Processes are important; however, they have to be light and efficient, so they're not regarded as an overhead to the development process and to avoid making engineers spend too much time on compliance activities.

2. Think long term and execute short term: the priority of all companies is shipping products on time, while also building and expanding their internal open-source compliance infrastructure. Therefore, expect to build your compliance infrastructure as you go, doing it the right way and keeping scalability in mind for future activities and products.

3. Establish a clean software baseline. This is usually an intensive activity over a period of time. The results of the initial compliance activities include a complete software inventory that identifies all open-source software in the baseline, a resolution of all issues related to mixing proprietary and open-source code, and a plan for fulfilling the license obligations for all the open-source software.

Building a Compliance Infrastructure

Here are the essential building blocks of an open-source compliance infrastructure required to enable open-source compliance efforts (Figure 2):

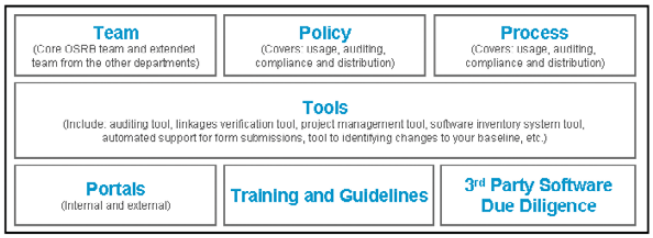


Figure 2. Open-Source Compliance Building Blocks

- Open-source review board (OSRB): comprises representatives from engineering, legal and open-source experts. The OSRB reviews requests for use, modification and distribution of open-source software and determines approval. In addition, the OSRB serves as a steering committee to define and manage your company's open-source strategy.
- Open-source compliance policy: typically covers usage, auditing and post-compliance activities, such as meeting license obligations and distribution of open-source software. Usual items mandated in a compliance policy are approval of OSRB for each piece of open-source software included in a product, ensuring that license obligations are fulfilled prior to customer receipt, mandatory source code audits, mandatory legal review and the process and mechanics of distribution.
- Open-source compliance process: the work flow through which a request to use an open-source component goes before receiving approval, including scanning code, identifying and resolving any flagged issues, legal review and the final decision. See HP's "FOSS Management Issues" article at **www.fsf.org/licensing/compliance** for an example of a compliance process.
- Compliance project management tool: some companies use bug-tracking tools that already were in place, and other companies rely on professional project management tools. Whatever your preference is, the tool should reflect the work flow of your compliance process, allowing you to move compliance tickets from one phase of the process to another, providing task and resource management, time tracking, e-mail notifications, project statistics and reporting.
- Open-source inventory management: it is critical to know what open-source software is included for each product, including version numbers, licensing information, compliance information and so on. Basically, you need to have a good inventory of all your open-source assets—a central repository for open-source software that has been approved for deployment. This inventory is handy for use by engineering, legal and OSRB.
- Open-source training: ensures that employees have a good understanding of your company's open-source policies and compliance practices, in addition to understanding some of the most-common open-source licenses. Some companies go one step further by mandating that engineers working with open-source software take open-source training and pass the evaluation.
- Open-source portals: companies usually maintain two open-source portals: an internal portal that houses the open-source policies, guidelines, documents, training and hosts a forum for discussions, announcements, sharing experiences and more; and an external portal that is a window to the world and the Open Source community and a place to post all the source code for open-source packages they use, in fulfillment of their license obligations with respect to distribution.
- Third-party software due diligence: you should examine software supplied to you by third parties carefully. If third-party software includes open-source software, ensure that license obligations are satisfied, because this is your responsibility as the distributor of a product that includes open-source software. You must know what goes into all of your product's software, including software provided by outside suppliers.

Who's Involved in Open-Source Compliance?

Several departments are involved in ensuring open-source compliance (Figure 3). Here's a generic breakdown of the different departments and their roles in achieving open-source compliance:

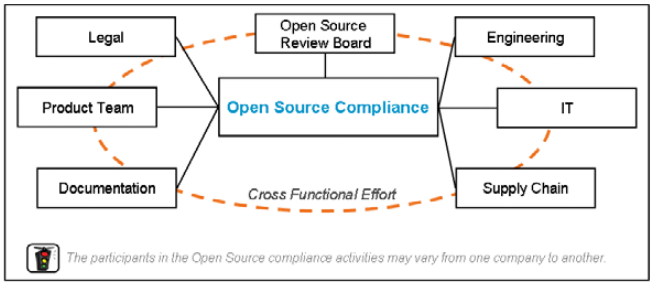


Figure 3. Teams Involved in Ensuring Open-Source Compliance

- Legal: advises on licensing conflicts, participates in OSRB reviews, and reviews and approves content of the open-source external portal.
- Engineering and product team: submits OSRB requests to use open-source software, participates in the OSRB reviews, responds promptly to questions asked by the compliance team, maintains a change log for all open-source software that will be made publicly available, prepares source code packages for distribution on the company's open-source public portal, integrates auditing and compliance as part of the software development process checkpoints, and takes available open-source training.
- OSRB team: drives and coordinates all open-source activities, including driving the open-source compliance process; performs due diligence on suppliers' use of open source; performs code inspections to ensure inclusion of open-source copyright notices, change logs and the like in source code comments; performs design reviews with the engineering team; compiles a list of obligations for all open-source software used in the product and passes it to appropriate departments for fulfillment; verifies fulfillment of obligations; offers open-source training to engineers; creates content for the internal and external open-source portals; and handles compliance inquiries.
- Documentation team: produces open-source license file and notices that will be placed in the product.
- Supply chain: mandates third-party software providers to disclose open-source software used in what is being delivered.
- IT: supports and maintains compliance infrastructure, including servers, tools, mailing lists and portals; and develops tools that help with compliance activities, such as linkage analysis.

Establishing Compliance Best Practices

The following compliance best practices fall under six major categories. Each of the categories represents a step in a typical compliance process (Figure 4).

1. Scanning Code The first step in the compliance process is usually scanning the source code, also sometimes called auditing the source code. Some common practices in this area include:

- Scanning everything—proprietary code, third-party software and even open-source software, because your team might have introduced modifications triggering the need for additional due diligence and additional obligations to fulfill.
- Scan early and often—scan as early in the development process and as often as possible to identify new packages entering your build.
- Scan newer versions of previously approved packages—in the event that a previously approved packaged was modified, you should rescan it to ensure that any code added to it does not have a conflicting license and that there are no additional obligations to meet.

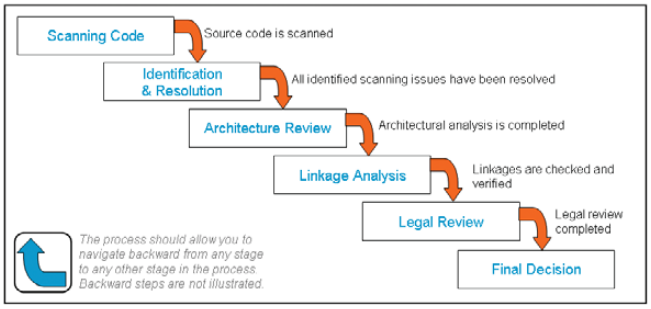


Figure 4. A Generic Open-Source Compliance Process

2. Identification and Resolution of Flagged Issues After scanning the source code, the scanning tool generates a report that includes a "Build of Material", an inventory of all the files in the source code package and their discovered licenses, in addition to flagging any possible licensing issues found and pinpointing the offending code. Here's what should happen next:

- Inspect and resolve each file or snippet flagged by the scanning tool.
- Identify whether your engineers made any code modifications. Ideally, you shouldn't rely on engineers to remember if they made code changes. You should rely on your build tools to be able to identify code changes, who made them and when.
- When in doubt of the scan results, discuss it with Engineering.
- If a GPL (or other) violation is found, you should report to Engineering and request a correction. Rescan the code after resolving the violation to ensure compliance.
- In preparation for legal review, attach to the compliance ticket all licensing information (COPYING, README, LICENSE files and so on) related to the open-source software in question.

3. Architecture Review The architecture review is an analysis of the interaction between the open-source code and your proprietary code. Typically, the architecture review is performed by examining an architectural diagram that identifies the following:

- Open-source components (used as is or modified).
- Proprietary components.
- Components' dependencies.
- Communication protocols.
- Linkages (dynamic and static).
- Components that live in kernel space vs. userspace.
- Shared header files.

The result of the architecture review is an analysis of the licensing obligations that may extend from the open-source components to the proprietary components.

4. Linkage Analysis The purpose of the linkage analysis is to find potentially problematic code combinations at the dynamic link level, such as dynamically linking a GPL library to proprietary source code component (Figure 5). The common practices in this area include:

- Performing dynamic linkage analysis for each package in the build.
- If a linkage conflict is identified, report to it Engineering to resolve.
- Redo the linkage analysis on the updated source code to verify that the code changes introduced by Engineering resolved the linkage issue.

As for static linkages, usually companies have policies that govern the use of static linkages, because it combines proprietary work with open-source libraries into one binary. These linkage cases are discussed and resolved on a case-by-case basis.

Figure 5 illustrates the difference between static and dynamic linking to highlight the importance of identifying how open-source license obligations can extend from the open-source components (libraries, in this example) to your proprietary code through the linking method.

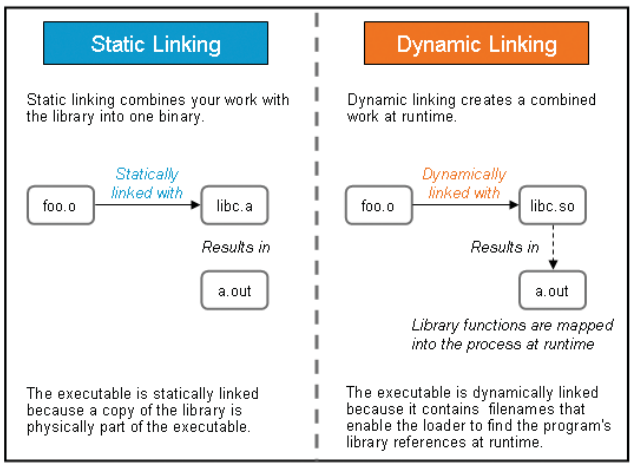


Figure 5. Static vs. Dynamic Linking

5. Legal Review The best practices of the legal review include:

- Review the report generated by the scanning tool attached to the compliance ticket.
- Review the license information provided in the compliance ticket.
- Review comments left in the compliance ticket by engineers and OSRB members.
- Flag any licensing conflict and reassign compliance ticket to Engineering to rework code if needed.
- Contact the open-source project when licensing information is not clear, not available or the code is licensed under more than one license with unclear terms/conditions.
- Decide on incoming and outgoing license(s)

Source Code Scanning Tools

There are commercial and open-source tools that offer the capabilities of scanning source code for potential open-source issues. Commercial tools include Protex from Black Duck Software, Inc. (**www.blackducksoftware.com/protex**) and Palamida Compliance Edition from Palamida (**www.palamida.com/products/complianceedition**). A popular open-source tool is FOSSology (**www.fossology.org**).

6. Final Review The final review is usually an OSRB face-to-face meeting during which open-source software packages are approved or denied usage. A good practice is to record the minutes of the meeting and the summary of the discussions leading to the decisions of approval or denial. This information can become very useful when you receive compliance inquiries. For approved open-source packages, the OSRB would then compile the list of obligations and pass it to appropriate departments for fulfillment.

Responding to Compliance Inquiries

This section presents guidelines to observe when dealing with compliance inquires. These guidelines aim to maintain a positive and collaborative attitude with the requester of compliance information while investigating the allegation and ensuring proper handling in case of license violation. Figure 6 illustrates the recommended steps to follow when dealing with open-source compliance inquiries.

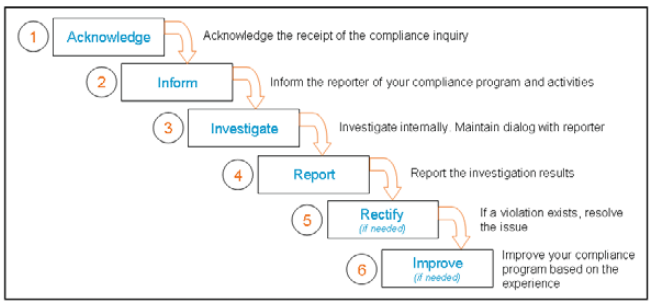


Figure 6. Handling an Open-Source Compliance Inquiry

Several companies received negative publicity and/or got sued because they either ignored requests to provide open-source compliance information, did not know how to handle compliance inquires, lacked or had a poor compliance program, or simply refused to cooperate, thinking it was not enforceable. By now, we know that none of these approaches is fruitful or beneficial to any of the parties involved. Therefore, as a general rule, companies should not ignore open-source compliance inquiries. Instead, they should acknowledge the receipt of the inquiry, inform the inquirer that they will look into it and provide a date when

Open-Source Compliance Insurance

In the past few years, some insurance companies started offering insurance services against the legal risks that can result from using open-source software. The insurance policy often is called open-source compliance insurance. The insurance policy (depending on the issuing company) offers coverage for monetary damages, including profit losses related to noncompliance with open-source software licenses and the cost of updating the offending code.

to expect a follow-up.

You should understand who the reporter is, the motivation and whether the accusation is accurate or even current. Furthermore, not every reporter understands licenses fully, and sometimes there may be mistakes in the submissions. Make sure you fully understand the inquiry and that you have all the necessary information to isolate the problem and investigate it internally. If that's not the case, ask the reporter to be specific and provide you with the missing details to start your investigation.

Keep an open dialog with the reporter and show that your company maintains rigid compliance practices. Highlighting your open-source compliance program and practices shows a good-faith effort toward compliance. Send updates of your internal investigation when they are available.

After concluding the internal investigation (within an acceptable time limit) through the review of the compliance due diligence completed for the specific software component (or product) in question, inform the reporter of the results.

If indeed there is a license violation as reported, it is your responsibility to resolve the issue with the reporter, while being collaborative and showing goodwill. You need to understand the obligations under the applicable license and show how you will meet the obligations and how soon.

Conclusion

This article provides an overview of open-source compliance, the challenges faced when establishing a compliance program, industry practices and recommendations on how to deal with compliance inquiries.

Open-source compliance is an essential part of the development process. Start with a simple, lightweight compliance process and practice and learn and adjust as you proceed. Look at common practices for inspiration, but most likely you will make adjustments to fit your specific company's needs.

If you use open-source software in your product(s), and you don't have a solid open-source compliance program, consider this article as a call to action.■

Ibrahim Haddad is Director of Open Source at Palm, Inc., and a Contributing Editor for *Linux Journal*.

Linux News and Headlines
Delivered To You
Linux Journal topical RSS feeds NOW AVAILABLE

http://www.linuxjournal.com/rss_feeds

SFLC's Practical Guide to GPL Compliance

On August 26, 2008, the Software Freedom Law Center (SFLC) published a guide on how to be compliant with the GNU General Public License (GPL) and related licenses. The guide focuses on avoiding compliance actions and minimizing the negative impact when enforcement actions occur. The guide is available at **www.softwarefreedom.org/resources**.

Resources

Free Software Foundation: **www.fsf.org**

Software Freedom Law Center: **www.softwarefreedom.org**

GNU Project: **www.gnu.org/licenses/gpl-violation.html**

IT is Continually Evolving, Be Sure to Keep Up.

Attend the most comprehensive IT events of the year, and gain the end-to-end views on enterprise technology that will help you keep up with the evolving needs of your data center.

Complimentary events for qualified attendees!

LINUXWORLD CONFERENCE & EXPO is now

OpenSource world

co-located with

NGDC
NEXT GENERATION DATA CENTER

and

CloudWorld

Three events. Tangible benefits. Immediate results.

From cost-effective, open source solutions and data center tools to cloud computing strategies, these events cover integrated, enterprise technologies aimed at increasing data center efficiency and reducing costs. The co-location of OpenSource World, NGDC and CloudWorld provides a unique value proposition that will maximize learning and use your time away from the office efficiently.

These events will enable you to:

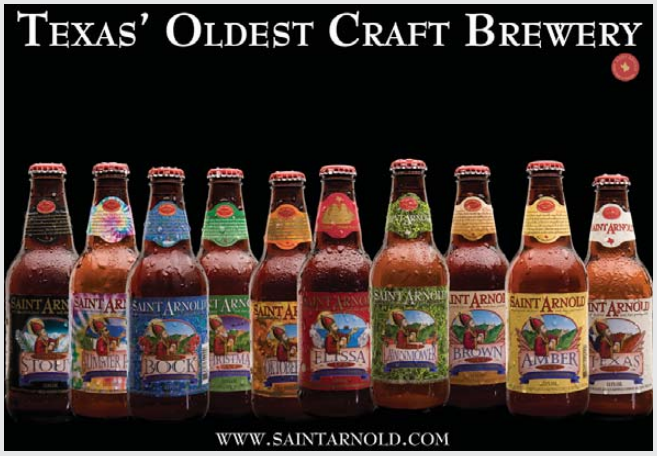
- Take home solutions and best practices that will immediately increase data center efficiency, while saving on IT costs.
- Get an in-depth look at technology trends and meet face-to-face with leading solutions providers.
- Meet with peers and share case studies for data center management, open source adoption, cloud computing implementation and much more.

REGISTER NOW to Qualify for Free Attendance!
www.opensourceworld.com
Attendance is limited to IT and business professionals who meet qualifying criteria.

For sponsorship opportunities, visit www.opensourceworld.com

AN IDG WORLD EXPO EVENT

AUGUST 12-13, 2009 | MOSCONE CENTER WEST | SAN FRANCISCO, CA | www.opensourceworld.com



STAY CONNECTED

- flickr.com/groups/linuxjournal/pool/
- twitter.com/linuxjournal
- linuxjournal.com/rss_feeds
- facebook.com/linuxjournal
- identi.ca/linuxjournal

EtherDrive®
The AFFORDABLE Network Storage
Fiber Channel speeds at Ethernet prices
CORAID
vmware® | technology alliance PARTNER
ESX 3.5 compatible EtherDrive® HBA

ARM9 System on Module
Internet Appliance Engine
SoM-9G20
• Atmel ARM9 400Mhz CPU
• 10/100 BaseT Ethernet
• SD/MMC Flash Card Interface
• 2 USB 2.0 Host Ports & 1 Device Port
• 6 Serial Ports, 2 SPIs & Audio Interface

The SoM-9G20 is the ideal processor engine for your next design. The System on Module (SoM) approach provides the flexibility of a fully customized product at a greatly reduced cost. Single unit pricing starts at $155.

EMAC Linux 2.6 Kernel
EMAC, inc.
EQUIPMENT MONITOR AND CONTROL
Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

Get your NEW *Linux Journal* Archive CD today featuring all issues from 1994 through 2008. Just $34.95.

www.linuxjournal.com/archivecd

YOUR TECHNOLOGY SHOULD WORK AS FAST AND SMART AS YOU DO.
And choosing Linux should never limit your technology options.
Parallels
Optimized Computing
Talk to the people who know Linux; talk to Parallels.
E: offer@parallels.com
P: 425.282.6448
www.parallels.com

LINUX JOURNAL WANTS YOU!
Free gift w/subscription. Sign-up today.
www.linuxjournal.com/subscribe

INNOVATION ON THE GO
ORDER YOUR BEAGLE BOARD FROM DIGIKEY.COM

AVAILABLE EXCLUSIVELY AT DIGI-KEY
beagleboard
only $149.00
LOW-COST, NO FAN, SINGLE-BOARD COMPUTER
Digi-Key® CORPORATION
www.digikey.com

EOF

Conferences: Pro & Un

What's the future of Linux conferences?

DOC SEARLS

LinuxWorld Expo is no more. In its place, IDG World Expo offers OpenSource World Conference & Expo, which is in the future as I write this and in the past as you read it. The conference happens (or happened) in early August in San Francisco, following O'Reilly's popular OSCON (Open Source CONference).

At the time of this writing, OpenSource World still appears at the domain LinuxWorldExpo.com. Adding irony to insult, it's an .aspx site, meaning it runs on a Microsoft Windows server. According to Netcraft, linuxworldexpo.com was first seen in December 1998, but has a hosting history on Windows that goes back only to June 2008. Meanwhile, 34 of O'Reilly's 36 sites are on Linux. The other two are on FreeBSD and Solaris. The oldest O'Reilly.com site on that list (one running Linux) dates back to March 1996.

Under "What We Do" on its home page, IDG World Expo (the organizer of LinuxWorld Expo and OpenSource World), pitches "Bringing You Face-to-Face with Decision-Makers You Want to Reach", adding "IDG is recognized worldwide as a leader in exhibition management, producing more than 750 globally branded conferences and events in 55 countries". O'Reilly Conferences' equivalent is, "O'Reilly conferences bring alpha geeks and forward-thinking business leaders together to shape the revolutionary ideas that spark new industries". Both are pitches to business. The difference is a company that talks the walk and one that walks the talk. Put more bluntly, one is driven by business and the other is driven by geekery.

IDG World Expo appears to keep no archives of its past conferences. O'Reilly does (**conferences.oreillynet.com/archive.csp**). So do we, because (being geeky like O'Reilly) we put pretty much everything we write on the Web, and then make sure it stays there. That's how I found Marjorie Richardson's coverage of the first LinuxWorld (**www.linuxjournal.com/article/3340**). She writes, "I spent a remarkable two days, March 2 and 3, in the San Jose Convention Center, and everyone who didn't go has been dropping by to find out about it. This was a major conference with more than just the usual suspects in attendance, and everyone had a big announcement." The second LinuxWorld followed in August at the same venue. About that one, CNN wrote, "'Linux is hot', said Madeline Schnapp, a product marketing manager for O'Reilly & Associates, which publishes books on open source. 'If you're anybody who is anybody in Linux, you have to show up at this show.'" What a difference a decade makes.

I once measured my approximate total conference attendance with schwag bags. Back in 2003–2004, we lived in a house where the garage had large utility drawers that I found perfect for filling with old promotional bags—very handy for when we needed to haul toys to the beach or replace the kid's latest lost knapsack. When we moved out, I needed to empty the drawer. So, for the fun of it, I counted how many unduplicated conference give-away bags I still had accumulated in there. The total came to more than a hundred.

As Yogi Berra famously said, "When you get to a fork in the road, take it." That's what I did with conferences. I didn't stop going to old-fashioned vendor-sponsored conferences, but I did start going to unconferences, starting with Dave Winer's first BloggerCon at Harvard Law School in 2003.

Although that marked my first encounter with the un side of conferencing, hackathons had been going on at least since OpenBSD geeks first used the term in 1999. Linux codefests and installfests have been happening since the early 1990s. Wikipedia credits the annual XML developers conference with using the term "unconference" first in 1998.

The spirit of the modern unconference, however, goes back to Open Space Technology, or OST, which was coined by Harrison Owen in 1985. The OpenSpaceWorld Wiki describes OST as "one way to enable all kinds of people, in any kind of organization, to create inspired meetings and events".

My own involvement with Open Space began when some geeks forked their conversation off Digital ID World, in October 2004. Meeting first on a Gillmor Gang podcast, the Identity Gang met next on a patio at the Fairmont Princess Hotel during one of Esther Dyson's (still much missed) PC Forum conferences, in March 2005. Later that year, we held the first Internet Identity Workshop (IIW), organized by Kaliya Hamlin, Phil Windley and myself, around Open Space principles.

Since then, we've been holding two IIWs per year, each organized as an Open Space event, and every time the results are dramatic. Development moves forward in tectonic leaps—especially where disagreements (or potential ones) are on the floor. We've used the same technique with two VRM (vendor relationship management, **cyber.law.harvard.edu/projectvrm/Main_Page**) workshops so far, with the same results. So I'm a believer.

But, that doesn't diminish my belief in the continuing need for traditional conferences. The market demands them, and many conference attendees don't want to participate in anything more demanding than visiting booths and sitting in keynote and breakout sessions. And, vendors still need places in the meet/meat world to show off their stuff.

At the time of this writing (June 2009), Wikipedia lists 32 Linux conferences, including OSCON and OpenSource World. Most are clearly by and for geek communities. Scoop Nisker says "If you don't like the news, go out and make some of your own." Same goes for conferences.■

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.

CHOOSE THE PERFECT HOSTING FIT FOR YOUR COMPANY

MH
MANAGED HOSTING

Fully customized and dedicated server configurations.

- Customization of configurations
- Full OS environment control
- Server, storage, security and network devices
- Dedicated Support Team built around you
- Managed by certified experts 24x7x365

CH
CLOUD HOSTING

Pay-as-you-grow solutions for sites, apps and email via The Rackspace Cloud.

- Load-and-go websites
- On-demand virtual Linux servers
- Fast, inexpensive web storage
- Easy deployment, scaling and online administration
- Round-the-clock management of infrastructure

EH
EMAIL HOSTING

Business-class email for companies large and small.

- Hosted Microsoft® Exchange
- POP/IMAP email with Outlook capabilities
- Mobile access
- Wireless device synchronization
- Managed by email specialists at all times

experience ***fanatical support***®

rackspace.com/linuxjournal • 888-571-8976

rackspace® HOSTING

redhat PREMIER HOSTING PARTNER Microsoft CISCO hp AMD vmware PCI

More GFLOPS, Less WATTS

Intel® Nehalem is here!

Higher Memory Bandwidth with DDR3 and QPI
Clusters and Servers Consume Less Power

Four Servers in a 2U Chassis with all Hot-Swap:

- 1200 Watt 1+1 supply, 12 Drives, and Server Modules!

FasTree™ ConnectX® QDR and DDR InfiniBand Switches and HCAs

Intel Professional Compiler Suite and Cluster Toolkit

- Version 11 with Nehalem Enhancements
- Academic Pricing Available

intel Channel Partner Premier Member 2009

Configure your next Cluster today!

www.microway.com/quickquote

GPU Computing

WhisperStation™

With 1 to 4 Tesla GPUs

Tesla C1060 GPU Performance:

- 1 TFLOPS per GPU
- 4 GB DDR3 per GPU
- 102 GB/Sec Bandwidth
- CUDA SDK

Run MATLAB® on Tesla with "Jacket"

Clusters With Tesla™

S1070 - 4 GPU Servers

- 36 GPUs + 36 CPUs + 24 TB in 24U
- 40 Gbps FasTree™ InfiniBand
- InfiniScope™ Network Monitoring

FREE 15-day trial available at **microway.com**

Microway®
Technology you can count on℠

508-746-7341
microway.com

GSA Schedule Contract Number: GS-35F-0431N