

Compojure | Sublime Text | R | GNU Awk 4.1 | Tizen | DNSSEC

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

AUGUST 2013 | ISSUE 232 | www.linuxjournal.com

SPONSORED BY

New
Relic®

PROGRAMMING

HOW TO:
DATA ANALYSIS
WITH R

**SUBLIME
TEXT**
EDIT CODE
LIKE A PRO

**GNU
AWK 4.1**
WHAT'S NEW



PLUS:
Leverage Your
Code Base with
**CODE
SEARCH**

A LOOK AT TIZEN
A NEW MOBILE-
OPTIMIZED OS

COMPOJURE
FOR WEB
DEVELOPMENT

SET UP DNSSEC
FOR YOUR
OWN ZONE

Some things are just better together.

When it comes to Application Performance Monitoring, New Relic's web & mobile solutions go together as well as:

Nerds & Data

Lando Calrissian & Capes

Hodor & Bran

R2-D2 & C-3PO

Tango & Cash

Boldly Going & the Starship Enterprise

Turner & Hooch

Samwise & Frodo

Dungeons & Dragons

Neo & Trinity

There's probably another great example, it's just not coming to us right now...

New Relic for web & mobile app monitoring, a match made in nerd heaven.



CONTENTS

AUGUST 2013
ISSUE 232

PROGRAMMING

FEATURES

60 Using the R Advanced Statistical Package

A look at data analysis
with R.

Mihalis Tsoukalos

74 Sublime Text: One Editor to Rule Them All?

The power of an IDE and
the speed of a regular
editor. We show you
what the buzz is about.

Ken Kinder

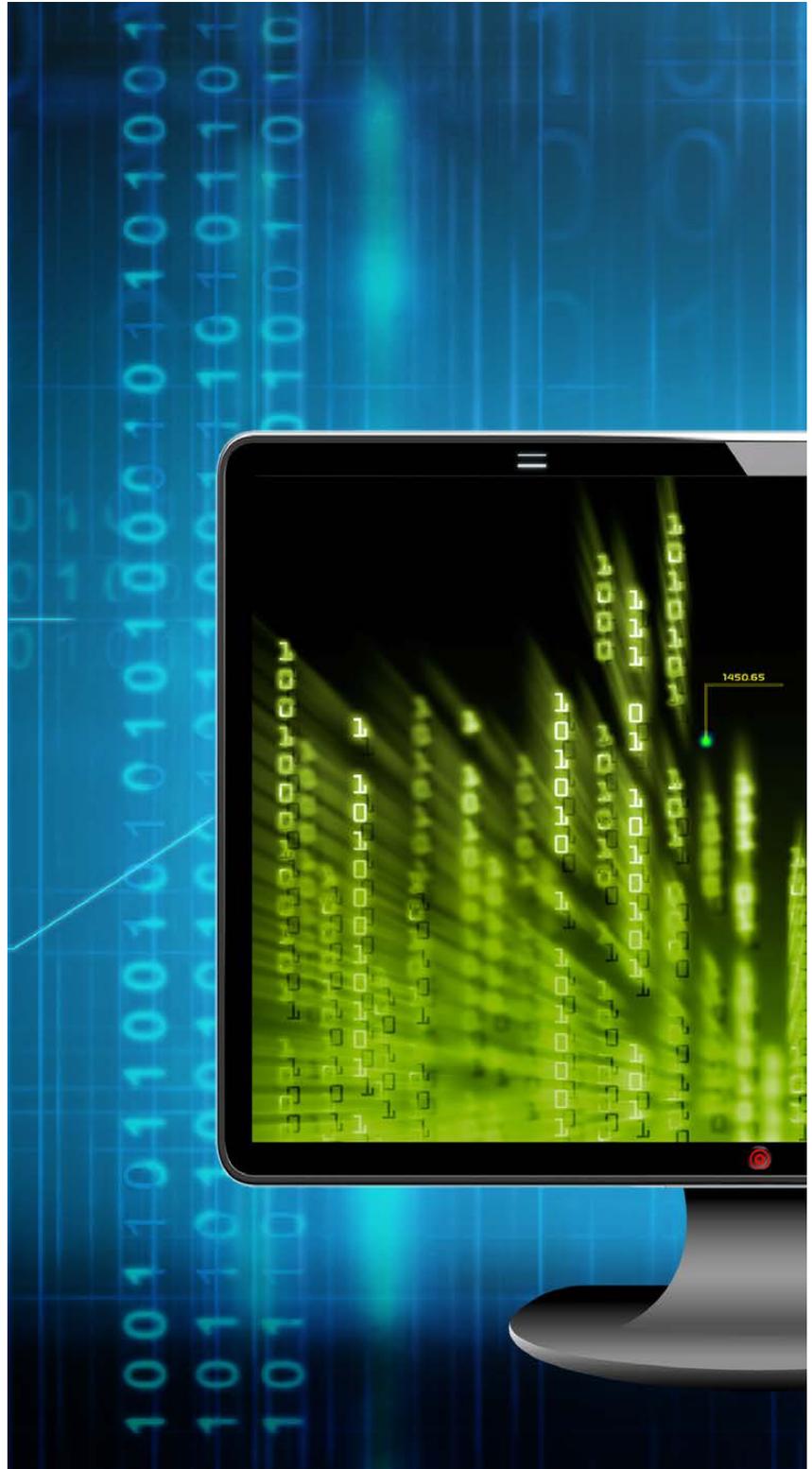
88 GNU Awk 4.1: Teaching an Old Bird Some New Tricks, Part II

The gawk developers
have not been idle.
Come see what's new!

Arnold Robbins

ON THE COVER

- How To: Data Analysis with R, p. 60
- Sublime Text: Edit Code Like a Pro, p. 74
- GNU Awk 4.1: What's New, p. 88
- A Look at Tizen, a New Mobile-Optimized OS, p. 104
- Compojure for Web Development, p. 30
- Set Up DNSSEC for Your Own Zone, p. 42
- Leverage Your Code Base with Code Search, p. 96



INDEPTH

96 Get More Juice out of Your Enterprise Code Base with Code Search

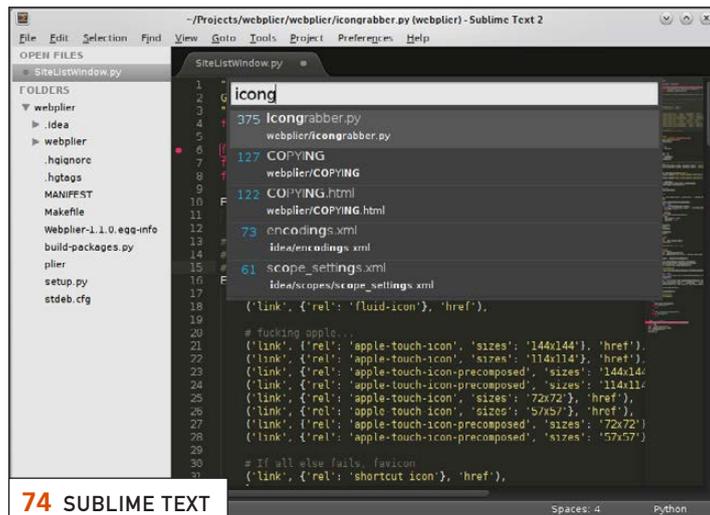
Gain access to the wealth of knowledge trapped inside your code base.

Sushil Krishna Bajracharya

104 Chances for a Tizen Smartphone Entry

Tizen's developers intend it to power a variety of devices including phones, tablets, vehicles and televisions.

Michael Schloh von Bennewitz



COLUMNS

30 Reuven M. Lerner's At the Forge Compojure

38 Dave Taylor's Work the Shell Web Administration Scripts

42 Kyle Rankin's Hack and / DNSSEC Part II: the Implementation

48 Shawn Powers' The Open-Source Classroom

Protect Your Ports with a Reverse Proxy

124 Doc Searls' EOF

Dear Hotels: Quit Being A-holes



DEPARTMENTS

8 Current_Issue.tar.gz

10 Letters

16 UPFRONT

28 Editors' Choice

56 New Products

129 Advertisers Index

Participate in the 2013 Readers' Choice Awards!



We're accepting nominations through August 18th. Actual voting is August 26th – September 22nd.

<http://www.linuxjournal.com/rc13>

LINUX JOURNAL™

Subscribe to
Linux Journal
Digital Edition
for only
\$2.45 an issue.



ENJOY:

Timely delivery

Off-line reading

Easy navigation

Phrase search
and highlighting

Ability to save, clip
and share articles

Embedded videos

Android & iOS apps,
desktop and
e-Reader versions

SUBSCRIBE TODAY!

LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@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
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors

Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan • Adam Monsen

Publisher Carlie Fairchild
publisher@linuxjournal.com

Director of Sales John Grogan
john@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

Editorial Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltys • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruiuzenga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

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
MAIL: PO Box 980985, Houston, TX 77098 USA

LINUX is a registered trademark of Linus Torvalds.

E5-2600

High Performance, High Density Servers for Data Center, Virtualization, & HPC

On-board 10 Gigabit Ethernet and Infiniband for greater throughput in less rack space

The Intel® Xeon® Processor E5-2600 family powers the highest-density servers iXsystems has to offer. The iXR-1204 +10G features dual onboard 10GigE + dual onboard 1GigE network controllers, up to 768GB of RAM and dual Intel® Xeon® E5-2600 family processors, freeing up critical expansion card space for application-specific hardware. The uncompromised performance and flexibility of the iXR-1204 +10G makes it suitable for clustering, high-traffic webservers, virtualization, and cloud computing applications - anywhere you need the most resources available.

For even greater performance density, the iXR-22X4IB squeezes four server nodes into two units of rack space, each with dual Intel® Xeon® E5-2600 Family Processors, up to 256GB of RAM, and an on-board Mellanox® ConnectX QDR 40Gbp/s Infiniband w/QSFP Connector. The iXR-22X4IB is perfect for high-powered computing, virtualization, or business intelligence applications that require the computing power of the Intel® Xeon® Processor E5-2600 family and the high throughput of Infiniband.

iXR-1204 +10G

- Dual Intel® Xeon® Processors E5-2600 Family
- Intel® X540 Dual-Port 10 Gigabit Ethernet Controllers
- Up to 16 Cores and 32 process threads
- Up to 768GB Main Memory
- 700W Redundant high-efficiency power supply

iXR-22X4IB

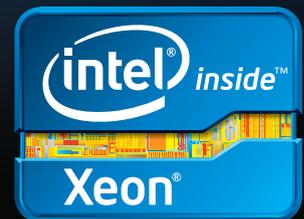
- Dual Intel® Xeon® Processors E5-2600 Family per node
- Mellanox® ConnectX QDR 40Gbp/s Infiniband w/QSFP Connector per node
- Four server nodes in 2U of rack space
- Up to 256GB Main Memory per server node
- Shared 1620W Redundant high-efficiency Platinum level (91%+) power supply



iXR-1204+10G: 10GbE On-Board



iXR-22X4IB



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



SHAWN POWERS

Building a Better Mouse (and Keyboard) Trap

I've mentioned in years past that my programming skills started with Pascal and ended with bubble sorting. My brain just doesn't seem wired to write code. Perhaps when actual brain-wiring moves from science fiction to the mainstream, I can upload something like, "How to Program for Neanderthals". Until that glorious cybernetic day, I'll continue to rely on the skills of others.

One of those folks is Reuven M. Lerner, who month after month gives us new insight into the programming world. This month, he expands on last issue's Clojure article and discusses Compojure, which allows us to connect to a PostgreSQL database. Our resident scripting expert, Dave Taylor, switches gears from *Cribbage* in this issue. Dave has been dealing with DDOS attacks of late, and he shares how he's using a script to detect the attacks on his server. Whether or not his attacker is an angry *Cribbage* player who can no longer beat his computer is still unknown!

Kyle Rankin introduced us to DNSSEC last month, and now that we have an understanding of how it works, he walks us through the process of implementation. By the time you reach the end of Kyle's article, you'll likely want to install DNSSEC for your domains. It's not the simplest technology to implement, but Kyle's teaching makes it feasible for us all. I follow Kyle's column with The Open-Source Classroom. Playing off last month's Tomcat installation, I walk through setting up a reverse proxy with Apache. Why add another layer of complexity to our server? Because running Tomcat and a Web server on the same machine usually means applications have nonstandard port numbers. With a reverse proxy, every application is a virtual host—no special port numbers to remember!

Although it sounds more like a pirate's greeting than a programming tool, Mihalis Tsoukalos shows us R this

month. R is a statistical package that offers some powerful tools, even for folks uncomfortable with mathematics and statistics. Whether you prefer to see the raw numbers or look at graphs generated to show your data, R is a tool everyone needing to sort through data should look into. When it comes to tools of the trade, nothing is more precious to a programmer than his or her text editor. Although vim is all I need as a system administrator, if you spend eight hours a day working with an editor, it should be one that makes your job easier. Ken Kinder introduces the Sublime Editor this month. It's a cross-platform, proprietary editor that offers a developer far more than syntax highlighting.

Arnold Robbins returns to a topic he covered a few years back by teaching us even more tricks with gawk. If awk and sed are your bread and butter, Arnold's article will feel like a home-cooked meal. Sushil Krishna Bajracharya follows up with a great article on using code search to utilize your enterprise's code base better. All too often we re-invent the wheel when it comes to programming because we don't know a solution already has been written! If you ever feel like you're re-inventing the wheel inside a wheel factory, Sushil's article is for you.

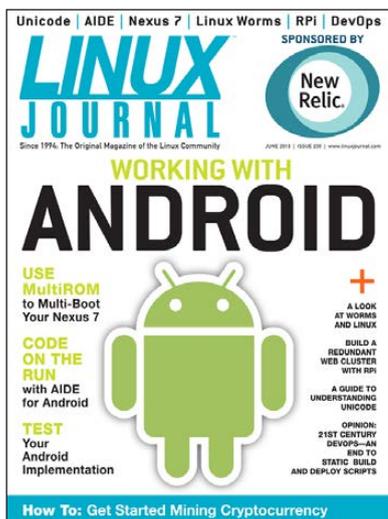
No programming issue would be complete without talking about programming for mobile devices. When

"Linux" and "Phones" are discussed, it seems that 99% of the time the discussion is about Android. Although awesome and powerful, Android isn't the only mobile OS leveraging Linux. Ubuntu has a mobile OS, Firefox has a mobile OS, and the world of Maemo/Moblin/Meego has transformed into Tizen. Michael Schloh Von Bennewitz explains all about the mobile platform you may not know about, but that has very deep roots in the mobile world. If you think competition in the mobile OS world is a good thing, check out Michael's article on Tizen, it's exciting stuff.

It's unlikely I'll be getting a cybernetic implant that allows me to jumpstart a programming career; however, issues like this month's always excite me. I enjoy reading about programming, and along with those focused articles, we have product announcements, tech tips and other goodies along the way. Oh, and to address the inevitable e-mail messages I'll get about being a test case for the new cybernetic learning tool you're writing? I'll wait for version 2.0, but thanks anyway. ■

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](https://www.freenode.net) IRC channel on Freenode.net.

letters



Worms and Linux

In Himanshu Arora's "Worms and Linux" article in the June 2013 issue, he mentions "Meanwhile, apart from

the Morris worm, very few worms have been directed toward Linux." The Morris worm was released in November 1988, three years prior to the initial Linux release in 1991.
—jetole

Himanshu Arora replies: *Thanks for your response. I acknowledge that you are correct, but what I actually meant here was *nix systems, although it would have been better if I had mentioned *nix explicitly.*

One Tail Just Isn't Enough

Regarding Shawn Powers' piece "One Tail Just Isn't Enough" in the June 2013 issue's Upfront section, "mutant felines" sounds neat, but the first thought in my head was...well, the first thought in my head that you can discuss in polite company would

be mutant canines. Specifically, I remember playing *Sonic The Hedgehog 2* on Sega Genesis when I was younger and we had the new sidekick "Tails". Tails is a two-tailed fox. Foxy!
—jetole

And, since you brought it up, I'll confess, the title occurred to me because I was reading Book 2 of the October Daye series by Seanan McQuire. In the book, the strength of a kitsune's magic is represented by the number of tails it has. (Yes, it sounds cheesy out of context, but it's a great series, I promise!)—Shawn Powers

GRASS GIS

Readers may well appreciate Joey Bernard's introduction to the GRASS GIS system in his "GIS with GRASS" article in the June 2013 Upfront section. GRASS is indeed the premier open-source GIS system. However, I think the article is remiss for Linux users in not noting that GRASS is inherently a CLI application. All of the individual commands are executable programs that capture and parse arguments to stdin. Accordingly, GRASS is fully scriptable in bash, tcsh, Python, Perl—almost anything. Personally, I never let GRASS open a GUI; I work in an xterm

(well, GnomeTerminal in my case).

The initial call to GRASS just sets a few environment variables (including putting the GRASS executables in your path of course) and doesn't even capture the terminal. You still can issue commands to your shell, piping stdin or stdout through GRASS, run vi or anything else. If your network connection speed is decent, you can run GRASS remotely through `ssh -X`.

In addition, a GIS is not just about the maps, it's a spatially enabled database. In GRASS, the back end can be SQLite, MySQL, PostgreSQL or others; you're not tied to a proprietary or minimally functional RDBM. Any changes to the database are immediately inherently implemented in the GIS, because it reads the same tables. GRASS spatial data input/output or exchange is largely managed by the incredible gdal/ogr libraries (<http://www.gdal.org>, once again CLI).

In my personal system, I run GRASS attached to Postgres, with the R statistical environment attached to the same tables in Postgres. If it's spatial or visual, I issue the command to GRASS; if it's query-based, I use psql to query

Postgres, and if it's computational, I use R to crunch the numbers. Often I dedicate one workspace to each of the three to maximize working space. It's a scriptable, seamless, open-source, highly functional system if you just use the CLI.

—Dave Roberts

Joey Bernard replies: I appreciate you bringing up all of the extra potential available in GRASS. I too use a terminal

7" Panel PC

PPC-E7+

- ARM9 400Mhz Fanless Processor
- Up to 1 GB Flash & 256 MB RAM
- 7" 800 x 480 TFT LED Backlit LCD
- Analog Resistive Touchscreen
- 10/100 Base-T Ethernet
- 3 RS232 & 1 RS232/422/485 Port
- 1 USB 2.0 (High Speed) Host port
- 1 USB 2.0 (High Speed) OTG port
- 2 Micro SD Flash Card Sockets
- SPI & I2C ports
- I2S Audio Interface w/ Line-in/out
- Operating Voltage of 12 to 26 Vdc
- Optional 2D Accelerated Video & Decoder
- Pricing start at \$550 for Qty 1



2.6 KERNEL



Designed and Manufactured in the USA the PPC-E7+ Compact Panel PC comes ready to run with the Operating System installed on Flash. Apply power and watch the Linux X Windows User Interface appear on the vivid 7" color LCD. Interact with the PPC-E7+ using the responsive integrated touch-screen. Everything works out of the box, allowing you to concentrate on your application, rather than building and configuring device drivers. Just Write-It and Run-It.

www.emacinc.com/panel_pc/ppc_e7+.htm

Since 1985

OVER

28

YEARS OF

SINGLE BOARD

SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

[LETTERS]

window as my main interface, but I try to cover enough of a scientific package to get people interested in playing with it. Once they have their feet wet, hopefully they go on to see how much more they can do with it. And thanks to letters like yours, people can find out how others use such powerful tools.

“They Said It” Mark Twain Quotation

The Mark Twain entry in the June 2013 “They Said It” column is actually a “They Didn’t Say It”. Although widely attributed to Twain on the Internet (usually in a political context), the quote is not his, and the you-are-on-your-own idea doesn’t fit the philosophy expressed in his many books and speeches. Like many of his age, Samuel Clemens was greatly concerned with inequality, corporate greed and power, corruption, and the plight of the downtrodden.

That small error aside, I enjoy your magazine...keep up the great work!

—Richard Merren

Thank you, Richard. Of all the articles and columns I write for Linux Journal, verifying quotes is actually the most difficult! I try to use them only if I can find a couple fairly reputable sources, but I do get some wrong—my apologies (and thank you for the correction).—Shawn Powers

RPi Issue

I really enjoyed the May 2013 issue on the Raspberry Pi. Prior to reading that issue, I just considered it a toy, soon to be another paperweight after playing with it. The articles opened my eyes. Now I have one myself, and it is configured as a print, Subversion and MySQL server for my local network. This allowed me to retire my old server, reducing power consumption by 75 watts—a nice benefit for such an inexpensive device. Keep up the good work.

—Craig

Awesome! I really had no desire to buy one either, but once Kyle Rankin told me about all he planned to do with his, I started to get jealous. Now, I’m really happy I bought some, because like you mentioned, they’re surprisingly powerful and useful for their size and electrical footprint.—Shawn Powers

Suggestions for the Browser Version of LJ

Since you went all-digital for *Linux Journal* issues, I’ve been quite happy with it. No more do I have to worry about lost, wrinkled, wet (or worse!) hard copy, and it doesn’t clutter up my living space. I have noticed that I mostly read *LJ* in my Web browser from my workstation or my laptop, at work,

and at home. I have used the .epub and .mobi versions on my Android phone, but that's relatively rare.

My first suggestion is that the browser version (and possibly the .mobi and .epub versions as well) use a cookie that stores my place in the magazine, so if I close my browser (Chromium being what it is), I don't lose the page I was on. As it stands, if I close Chromium, I have to page through the first part of the magazine to find my spot. This is annoying.

The other suggestion I have is about keyboard navigation. I discovered the one-page/two-page layout, and at least on this laptop, the one-page layout is better for me. If I use the arrow keys or the Page Up/Page Down keys to navigate to the next (or previous) page, the scroll slider should jump to the top of the resulting page. The way it works now, if I use the down arrow or Page Down to scroll to the next page, it brings me to the bottom of the next page, rather than the top. It seems the vertical scrollbar for the page keeps its state from the previous page. This forces me to scroll back to the top while trying to avoid going to the previous page. I use a combination of Page Up and up arrow keystrokes to bring my view to the

top, and hope I don't go too far.

I would think loading each new page at the top rather than the previous state of the vertical scrollbar would satisfy this request. Maybe more ideal still, if down/Page Down is used to navigate to the next page, reset the page scroll slider to the top of the next page. If up/Page Up is used to navigate to the previous page, reset the page scroll slider to the bottom.

Other than those annoyances, I'm very happy with the the digital editions of *LJ*.

—Trey Blancher

Thanks Trey. There is another company we hire to do the Web hosting, so we'll be sure to pass your suggestions on to it. I think it's great to have so many versions of the magazine available, because depending on where I am, I can view the content in multiple ways. Thanks again for your recommendations. Without feedback, things will never improve!—Shawn Powers

Script to Cut PDF Sections

Recently, somewhere in a *Linux Journal* issue, I saw a one-line bash command that would cut out a section of a PDF and save that section as a new PDF. I apologize, but I have misplaced the author of that valuable one-liner.

[LETTERS]

Anyway, finding this useful, I created a bash script that accomplishes this, and it accepts command-line arguments or GUI via the zenity package. The source is available at <http://pastebin.com/JYdnusJt>.

—celem

Thanks Celem!—Ed.

Photo of the Month

Here is a photo of Tux with my daughter. Tux was 3-D printed for me by shapeways, and it is in place of the standard VW logo that usually resides in this location on the hood of my VW Golf.

—Darryl Moore



WRITE LJ A LETTER We love hearing from our readers. Please send us your comments and feedback via <http://www.linuxjournal.com/contact>.

PHOTO OF THE MONTH

Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

LINUX JOURNAL

At Your Service

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: <http://www.linuxjournal.com/subs>. E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE: Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at <http://www.linuxjournal.com/digital>.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at <http://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: <http://www.linuxjournal.com/author>.

FREE e-NEWSLETTERS: *Linux Journal* editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on <http://www.linuxjournal.com>. Subscribe for free today: <http://www.linuxjournal.com/enewsletters>.

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: <http://www.linuxjournal.com/advertising>. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

1&1 DYNAMIC CLOUD SERVER PRICE CONTROL

Server Configuration

Processor Cores
\$0.02/CPU/hour
1 2 3 4 5 6 7 8

RAM (GB)
GB - \$0.02/GB/hour
1 2 4 8 10 12 16 20 32

Hard Disk Space (GB)
GB - \$0.02/GB/hour
100 200 400 600 800

Operating System

- Linux
 - Standard
 - Business
- Windows
 - Standard
 - Business

NEW!
Snapshot included

AS LOW AS
\$0.06
PER HOUR*
PLUS \$60 OFF
FIRST MONTH

✓ COMPLETE COST CONTROL

- Full transparency with accurate hourly billing
- Parallels® Plesk Panel 11 included, with unlimited domains

✓ FULL ROOT ACCESS

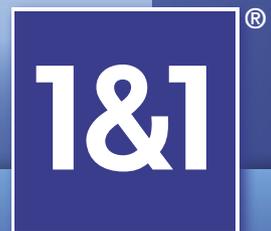
- The complete functionality of a root server with dedicated resources

✓ MAXIMUM FLEXIBILITY

- Configure the Processor Cores, RAM and Hard Disk Space
- Add up to 99 virtual machines

✓ FAIL-SAFE SECURITY

- Redundant storage and mirrored processing units reliably protect your server



DOMAINS | E-MAIL | WEB HOSTING | eCOMMERCE | SERVERS

Call 1 (877) 461-2631 or buy online

1and1.com

* Other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Customer is billed monthly for minimum configuration (\$0.06/ hour * 720 hours = \$43.20/ month minimum). A \$60 credit (valid only for the first month) will be applied to your first month of service. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet, all other trademarks are the property of their respective owners. © 2013 1&1 Internet. All rights reserved.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Union filesystems don't have much luck with the kernel development process. **Miklos Szeredi** recently tried to get **OverlayFS** into the main tree, but he ran into a wall in the form of **Al Viro**. **Linus Torvalds** initially responded to Miklos' request with, "I think we should just do it. It's in use, it's pretty small, and the other alternatives are worse." But when Al started reviewing the code, he found that the underlying filesystem operations were simply way too fragile to support users. Even simple operations like deleting a directory tree would be fraught with messy details that could leave the whole filesystem in an inconsistent state in the event of any interruption. In the end, he couldn't let the code pass through the gates.

Daniel Phillips made some extravagant claims about **Tux3** performance recently, and he got slapped around by some kernel folks for it. Apparently, Tux3 had outperformed **tmpFS** on some particular benchmarks, and Daniel was crowing about it on the mailing

list. But after folks like **Dave Chinner** took a look at the actual numbers, it became clear that the benchmark was unreproducible, and had been specifically engineered to measure only the asynchronous front end of Tux3, so that all the time-consuming hard work behind the scenes never actually was included in the benchmark. There was some grumbling from kernel developers about this, while Daniel argued that the benchmark tested only portions of the code that already had been implemented and that other tests would be done as more of the code was written. Clearly, there are two sides to the story. But as Dave Chinner put it, benchmarks should at least include enough information to reproduce the results.

How should Linux handle **empty symlinks**? At the moment, Linux doesn't allow users to create them, so you might think there's no problem—if they can't exist, there's no need to handle them one way or another. But, nothing prevents someone from mounting a filesystem that was created

on an operating system that does allow empty symlinks. So evidently, there really is a need to handle them properly if they ever appear.

As it turns out, Linux's current behavior is not very well known regarding this issue. **Pavel Machek** started exploring the various ins and outs of it, but the full scope and nuance may take a while to dig out. But, thanks to **Eric Blake's** cogent arguing, it's clear that something does need to be done. This is a case of **POSIX noncompliance** that actually may burn some people, as opposed to the cases of POSIX noncompliance that Linus Torvalds doesn't care about at all, in any way whatsoever. As far as Linus is concerned, if it doesn't hurt anyone, it's not a bug. If there's a way to improve on POSIX, then POSIX is the bug. But this time, it may be that POSIX isn't the bug, and the bug does bite.

Once in a while the **GPL v2** becomes the topic of debate. This time, **Luke Leighton** posted to the mailing list, saying that he wanted all his kernel contributions to be dual-licensed under the GPL v2 and the **GPL v3** (and all subsequent versions). But, **Cole Johnson** and **Theodore Ts'o** pointed out that Linus Torvalds, and many other top kernel people, very vocally had rejected the GPL v3 for the Linux kernel. Theodore said, "the

anti-Tivoization clause in GPLv3 is totally unacceptable, and so many of us have stated unequivocally that our code will be released under a GPLv2-only license. This means that GPLv3-only code is always going to be incompatible with code released as part of the Linux kernel, because substantial parts of the kernel have and will be available only under a GPLv2-only license."

At one point in the conversation, **Rob Landley** said that the loss of compatibilities between the GPL v2 and v3 had ruined "copyleft". He said, "These days the GPL largely serves to *prevent* code re-use, and people have responded to the perceived problems with 'GPL-next' initiatives where they fragment copyleft further with Affero variants, by using creative commons on code, and so on. But copyleft only ever worked as one big universal license, and now it doesn't."

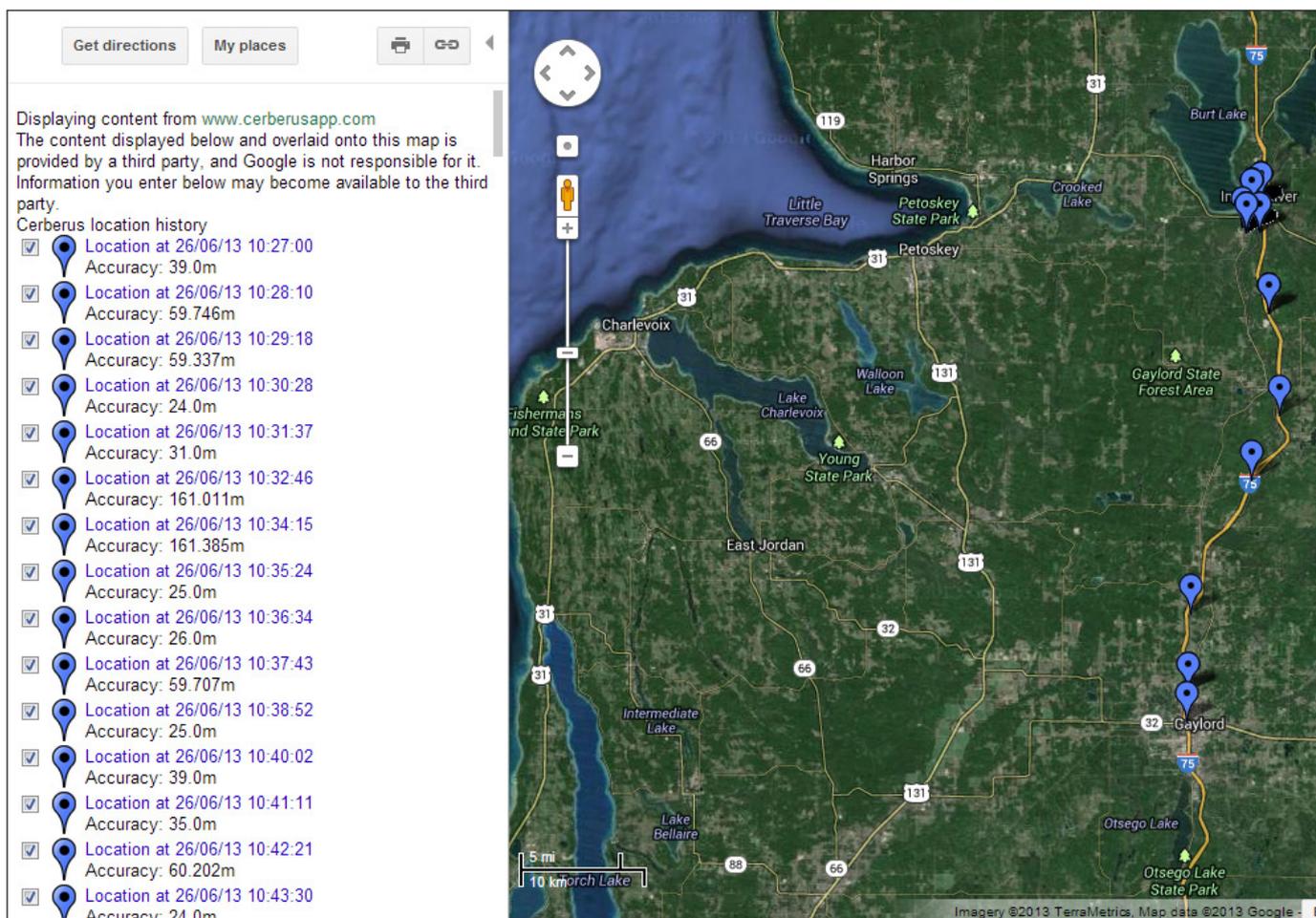
He added, "In the absence of a universal receiver, most developers have switched to universal donor licenses: MIT/BSD or even public domain. Yes, 'most': the most common license on GitHub is 'no license specified', and that's not just ignorance, that's napster-style civil disobedience from a generation of coders who lump copyright in with software patents and consider it all 'too dumb to live'."

A bleak assessment.—**ZACK BROWN**

Android Candy: Hire a Cerberus to Find Your Phone

In a recent career shift, I went from an employer who provided me an iPhone to one who provides me with an Android (Galaxy S4 to be specific). Although I was happy to move to a Linux-based handset, I

was concerned about replacing the “Find My iPhone” capability that Apple provides. Not only does my family use it to keep track of each other, but we also relied on it when a phone was misplaced. Does the



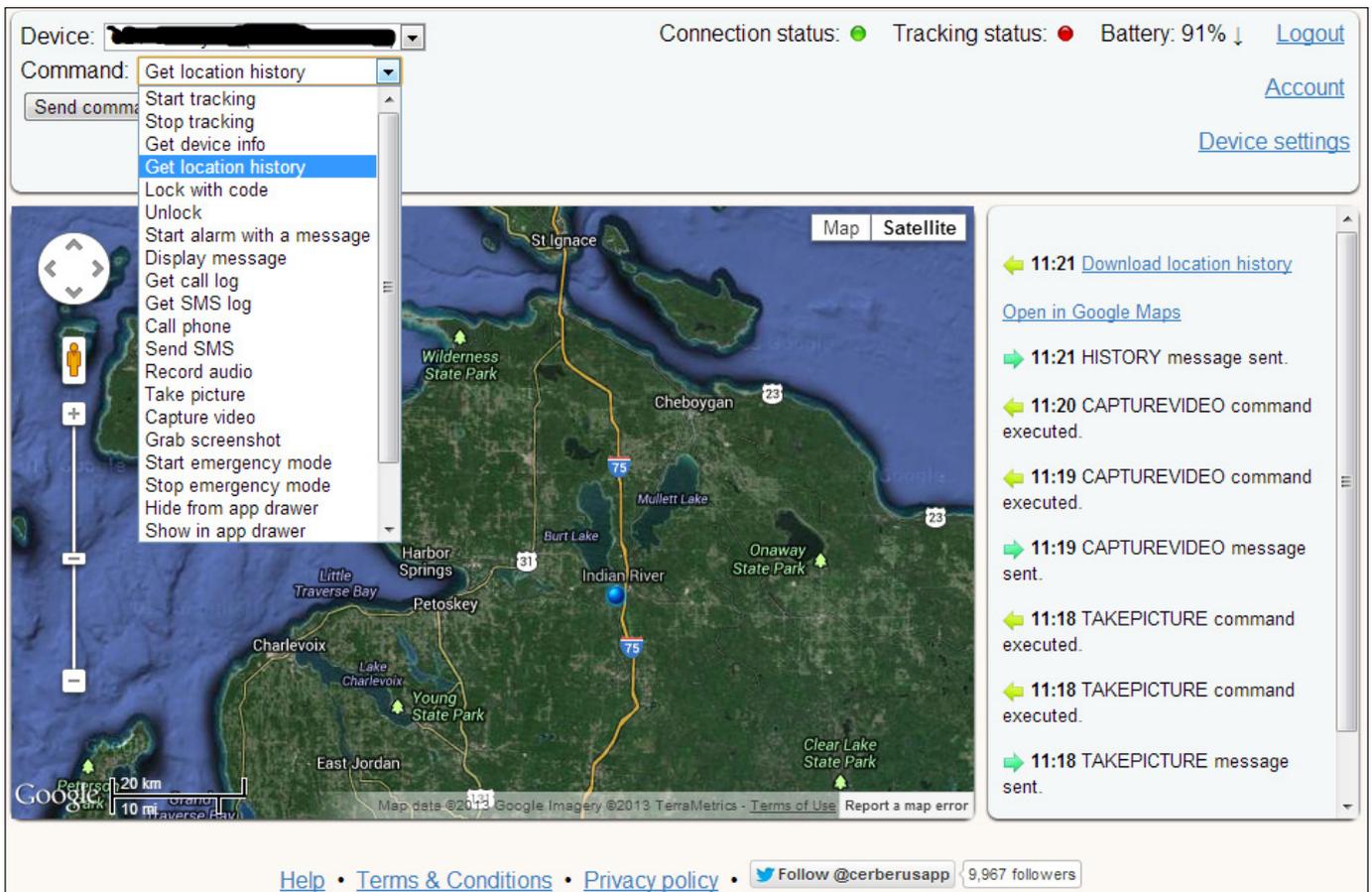


Figure 2. Cerberus' Features

Google Play store offer anything comparable? Um, yes.

Cerberus is a \$4 application (with a generous trial period so you can check it out) that blows Apple's "Find My iPhone" out of the water. Not only can it track down a phone, but it also keeps a history of where the phone has been (Figure 1), takes photos and videos, and yes, sets off an alarm to find your misplaced phone.

I was worried Cerberus might cause unusually high battery usage

due to its regular GPS pings, but I haven't noticed any difference at all. Plus, with all its features (Figure 2), I'd be willing to sacrifice a little battery life. Thankfully, I get the best of both worlds!

If you are switching from an iPhone to an Android device, or if you've been using Android for a while but haven't installed a security device, I urge you to try Cerberus (<http://www.cerberusapp.com>). It's awesome!

—**SHAWN POWERS**

Advanced OpenMP

Because this issue's theme is programming, I thought I should cover some of the more-advanced features available in OpenMP. Several issues ago, I looked at the basics of using OpenMP (<http://www.linuxjournal.com/content/big-box-science>), so you may want go back and review that article. In scientific programming, the basics tend to be the limit of how people use OpenMP, but there is so much more available—and, these other features are useful for so much more than just scientific computing. So, in this article, I delve into other by-waters that never seem to be covered when looking at OpenMP programming. Who knows, you may even replace POSIX threads with OpenMP.

First, let me quickly review a little bit of the basics of OpenMP. All of the examples below are done in C. If you remember, OpenMP is defined as a set of instructions to the compiler. This means you need a compiler that supports OpenMP. The instructions to the compiler are given through pragmas. These pragmas are defined such that they appear as comments to a compiler that doesn't support OpenMP.

The most typical construct is to use a for loop. Say you want to create an array of the sines of the integers from 1 to some maximum value. It would look like this:

```
#pragma omp parallel for
for (i=0; i<max; i++) {
    a[i] = sin(i);
}
```

Then you would compile this with GCC by using the `-fopenmp` flag. Although this works great for problems that naturally form themselves into algorithms around for loops, this is far from the majority of solution schemes. In most cases, you need to be more flexible in your program design to handle more complicated parallel algorithms. To do this in OpenMP, enter the constructs of sections and tasks. With these, you should be able to do almost anything you would do with POSIX threads.

First, let's look at sections. In the OpenMP specification, sections are defined as sequential blocks of code that can be run in parallel. You define them with a nested structure of pragma statements. The outer-most

layer is the pragma:

```
#pragma omp parallel sections
{
    ...commands...
}
```

Remember that pragmas apply only to the next code block in C. Most simply, this means the next line of code. If you need to use more than one line, you need to wrap them in curly braces, as shown above. This pragma forks off a number of new threads to handle the parallelized code. The number of threads that are created depends on what you set in the environment variable `OMP_NUM_THREADS`. So, if you want to use four threads, you would execute the following at the command line before running your program:

```
export OMP_NUM_THREADS=4
```

Inside the sections region, you need to define a series of individual section regions. Each of these is defined by:

```
#pragma omp section
{
    ...commands...
}
```

This should look familiar to

anyone who has used MPI before. What you end up with is a series of independent blocks of code that can be run in parallel. Say you defined four threads to be used for your program. This means you can have up to four section regions running in parallel. If you have more than four defined in your code, OpenMP will manage running them as quickly as possible, farming remaining section regions out to the running threads as soon as they become free.

As a more complete example, let's say you have an array of numbers and you want to find the sine, cosine and tangents of the values stored there. You could create three section regions to do all three steps in parallel:

```
#pragma omp parallel sections
{
    #pragma omp section
    for (i=0; i<max; i++) {
        sines[i] = sin(A[i]);
    }
    #pragma omp section
    for (j=0; j<max; j++) {
        cosines[j] = cos(A[j]);
    }
    #pragma omp section
    for (k=0; k<max; k++) {
        tangents[k] = tan(A[k]);
    }
}
```

[UPFRONT]

In this case, each of the section regions has a single code block defined by the for loop. Therefore, you don't need to wrap them in curly braces. You also should have noticed that each for loop uses a separate loop index variable. Remember that OpenMP is a shared memory parallel programming model, so all threads can see, and write to, all global variables. So if you use variables that are created outside the parallel region, you need to avoid multiple threads writing to the same variable. If this does happen, it's called a race condition. It might also be called the bane of the parallel programmer.

The second construct I want to look at in this article is the task. Tasks in OpenMP are even more unstructured than sections. Section regions need to be grouped together into a single sections region, and this entire region gets parallelized. With tasks, they are dumped onto a queue, ready to run as soon as possible. Defining a task is simple:

```
#pragma omp task
{
...commands...
}
```

In your code, you would create a

general parallel region with the pragma:

```
#pragma omp parallel
```

This pragma forks off the number of threads that you set in the `OMP_NUM_THREADS` environment variable. These threads form a pool that is available to be used by other parallel constructs.

Now, when you create a new task, one of three things might happen. The first is that there is a free thread from the pool. In this case, OpenMP will have that free thread run the code in the task construct. The second and third cases are that there are no free threads available. In these cases, the task may end up being scheduled to run by the originating thread, or it may end up being queued up to run as soon as a thread becomes free.

So, let's say you have a function (called `func`) that you want to call with five different parameters, such that they are independent, and you want to have them run in parallel. You can do this with the following:

```
#pragma omp parallel
{
for (i=1; i<6; i++) {
#pragma omp task
    func(i);
}
```

```
}
}
```

This will create a thread pool, and then loop through the for loop and create five tasks to farm out to the thread pool. One cool thing about tasks is that you have a bit more control over how they are scheduled. If you reach a point in your task where you can go to sleep for a while, you actually can tell OpenMP to do that. You can use the pragma:

```
#pragma omp taskyield
```

When the currently running thread reaches this point in your code, it will stop and check the task queue to see if there are any waiting to run. If so, it will go ahead and start one of those and put your current task to sleep. When the new task finishes, the suspended task gets picked up and resumes where it left off.

Hopefully, seeing some of the less-common constructs has inspired you to go and check out what other techniques you might be missing from your repertoire. Most parallel frameworks allow you to do most techniques. But each one, for historical reasons, has tended to be used for only one subset of techniques, even though there are constructs available that hardly ever are used. For shared memory programming, the constructs I cover here allow you to do many of the things you can do with POSIX threads without the programming overhead. You just have to trade some of the flexibility you get with POSIX threads.

—**JOEY BERNARD**

They Said It

Life is a great big canvas; throw all the paint on it you can.

—*Danny Kaye*

To achieve great things we must live as though we were never going to die.

—*Marquis de Vauvenargues*

It's choice—not chance—that determines your destiny.

—*Jean Nidetch*

Love all, trust a few. Do wrong to none.

—*William Shakespeare*

It is a mistake to try to look too far ahead. The chain of destiny can only be grasped one link at a time.

—*Sir Winston Churchill*

Non-Linux FOSS: Rearrange Your Furniture, Not Your Spine

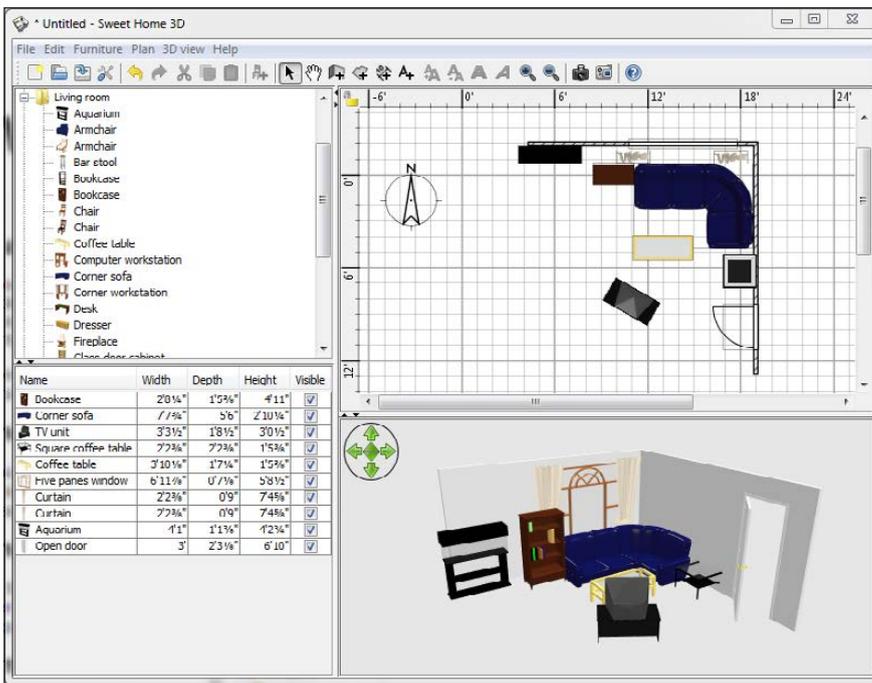


Figure 1. Living Room Design

My family is in the middle of moving from one house to another. Part of that move involves arranging furniture. I'll be honest, I can move a couch across a room only so many times before I start to think perhaps there's a better way. Thankfully, there is.

Although several 3-D house-modeling packages exist, and a couple are even on-line, nothing seems to work quite as simply as Sweet Home 3D. It's both a 3-D and

2-D layout tool, and it comes with a wide variety of pre-made furniture and window/door graphics to get you started. I was able to design a rudimentary living room in about two minutes (Figure 1), and that included installation time! Sweet Home 3D is an open-source Java application that comes with a nice Windows executable installer.

You might be thinking, if it's Java, won't it run on other platforms too? Well, yes, of course! It might not be as simple as the Windows executable installer to use it on OS X or Linux, but it's Java, so it's cross-platform-compatible. If you need to design a layout for your house, but don't want to haul furniture around to see what it looks like, I highly recommend Sweet Home 3D (<http://www.sweethome3d.com>).

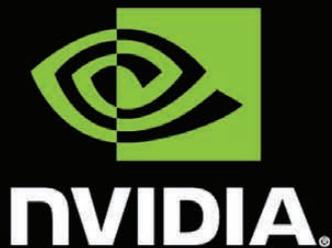
—SHAWN POWERS

Accelerate Your App

Are you tired of watching a sad workstation chug through your data?

Sign up for a test drive on one of our GPU server solutions today. See how you can accelerate your code or applications with parallel processing on NVIDIA® Tesla® K20 GPUs.

<http://www.siliconmechanics.com/testdrive>



Window Maker, the Unity for Old Guys?

As I was diving back into Window Maker for this article, it occurred to me that the desktop manager I used for years with Debian is disturbingly similar to the Unity Desktop. It's been clear since its inception that I am not

a fan of Ubuntu's new Unity interface, yet it's odd that for years I loved Window Maker, which seems fairly similar, at least visually.

After a little bit of usage, however, I quickly remembered why Window

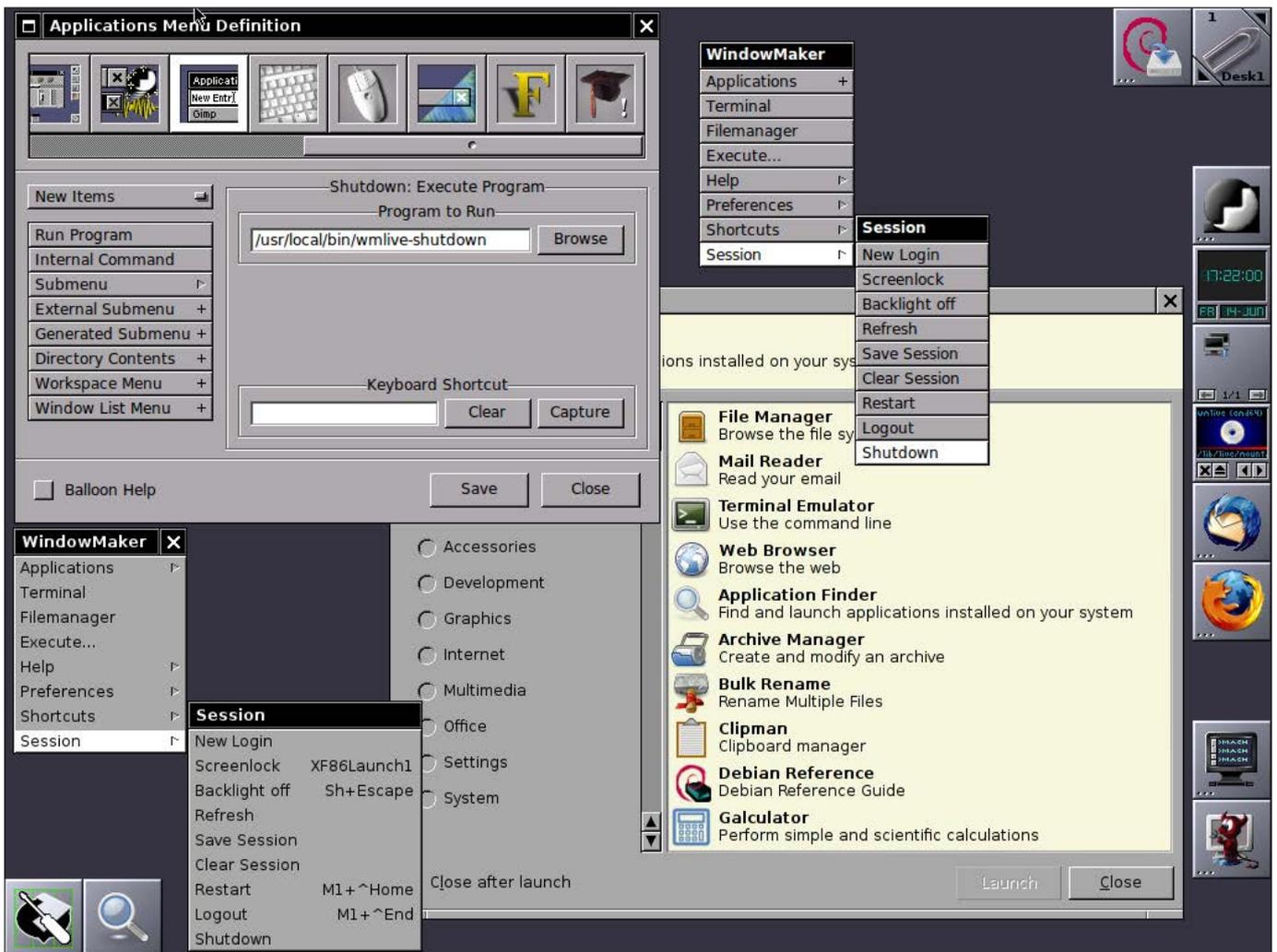


Figure 1. Window Maker is very customizable (screenshot from <http://wmlive.sourceforge.net>).

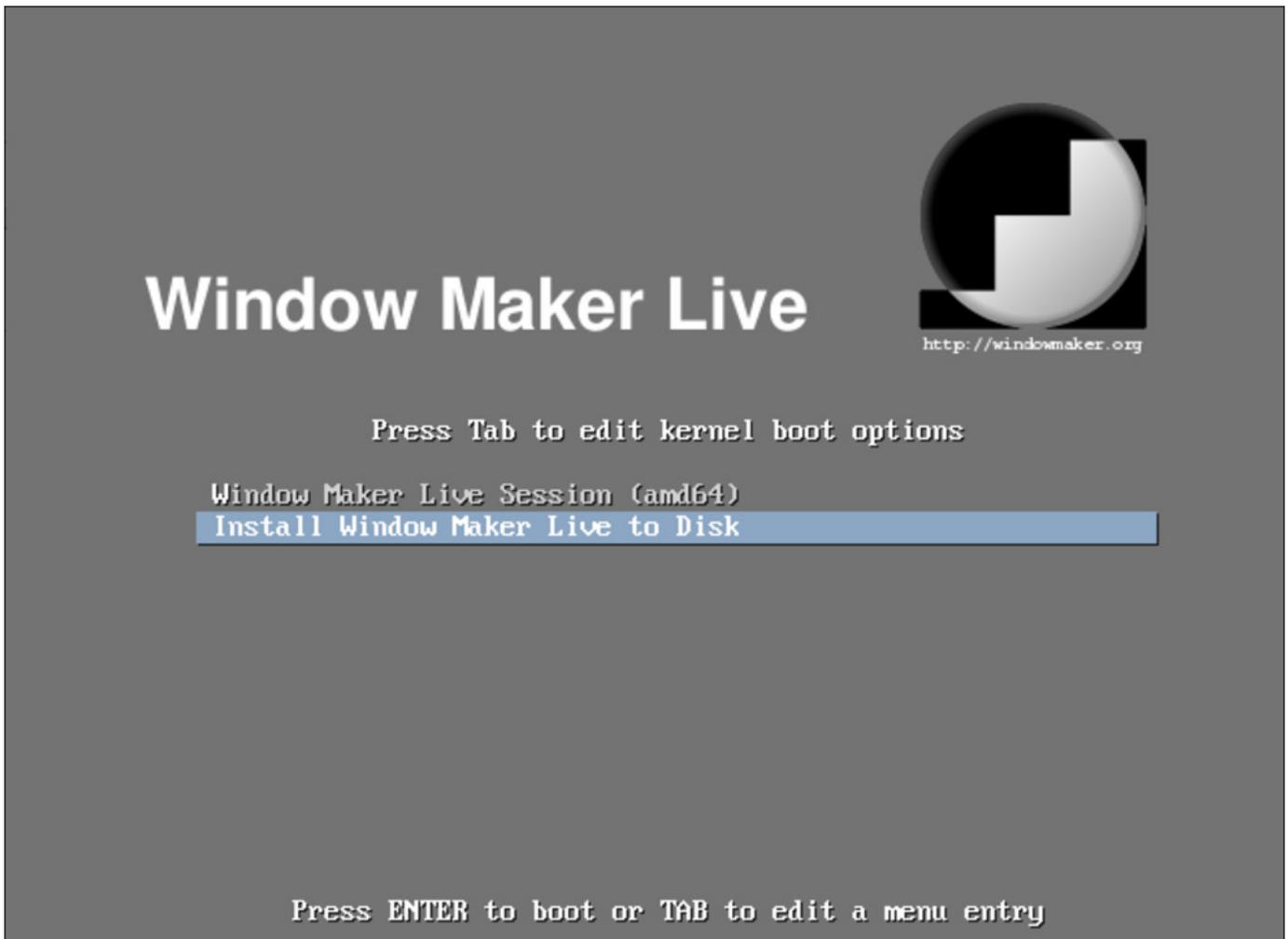


Figure 2. Window Maker installs the full Debian system directly from CD (screenshot from <http://wmlive.sourceforge.net>).

Maker was my desktop of choice for many years. Yes, it has the “side dock” look and feel, but it’s far, *far* more customizable (Figure 1). The dockapps can launch applications, certainly, but they also can be applications (widgets?) themselves, providing interaction and feedback instead of just eye candy.

The Window Maker Live CD actually is a great way to install Debian too. If you’ve never experienced Window

Maker firsthand, I urge you to download the ISO file from <http://wmlive.sourceforge.net>, and give the live CD a try. If you like it, it’s certainly easy to install the full Debian system directly from the CD (Figure 2). Window Maker is a low-resource, awesome desktop environment that’s worth checking out, at least for a weekend project.

—**SHAWN POWERS**



Songbird Becomes... Nightingale!

Several years back, Songbird was going to be the newest, coolest, most-awesome music player ever to grace the Linux desktop. Then things happened, as they often do, and Linux support for Songbird was discontinued. I've been searching for a favorite music player for

years, and although plenty of really nice software packages exist, I generally fall back to XMMS for playing music—until now.

Nightingale is truly everything I want in a music player. It is simple, yet powerful. The default install makes listening to music an

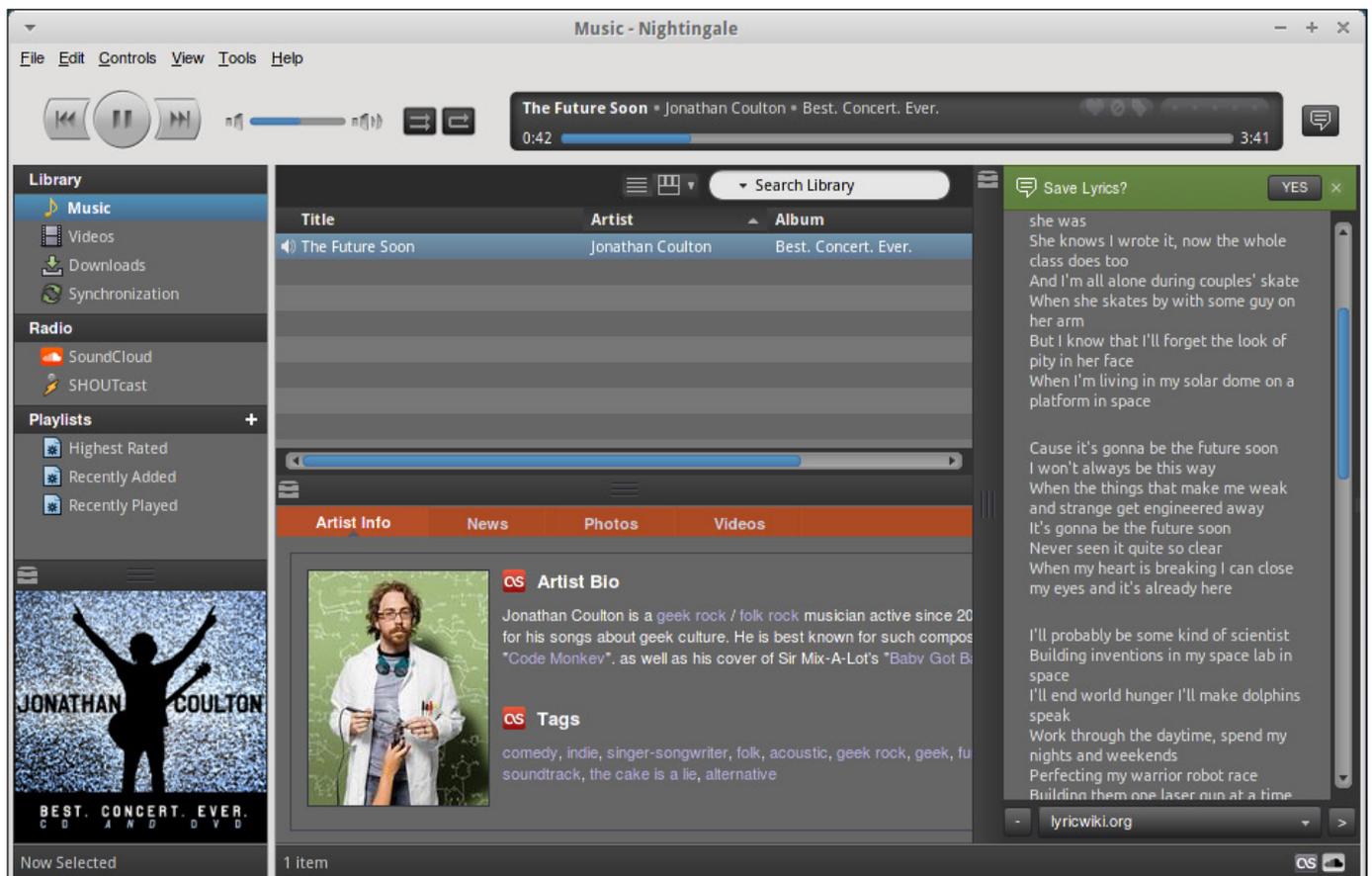


Figure 1. Playing a Song Shows the Lyrics and Artist Info

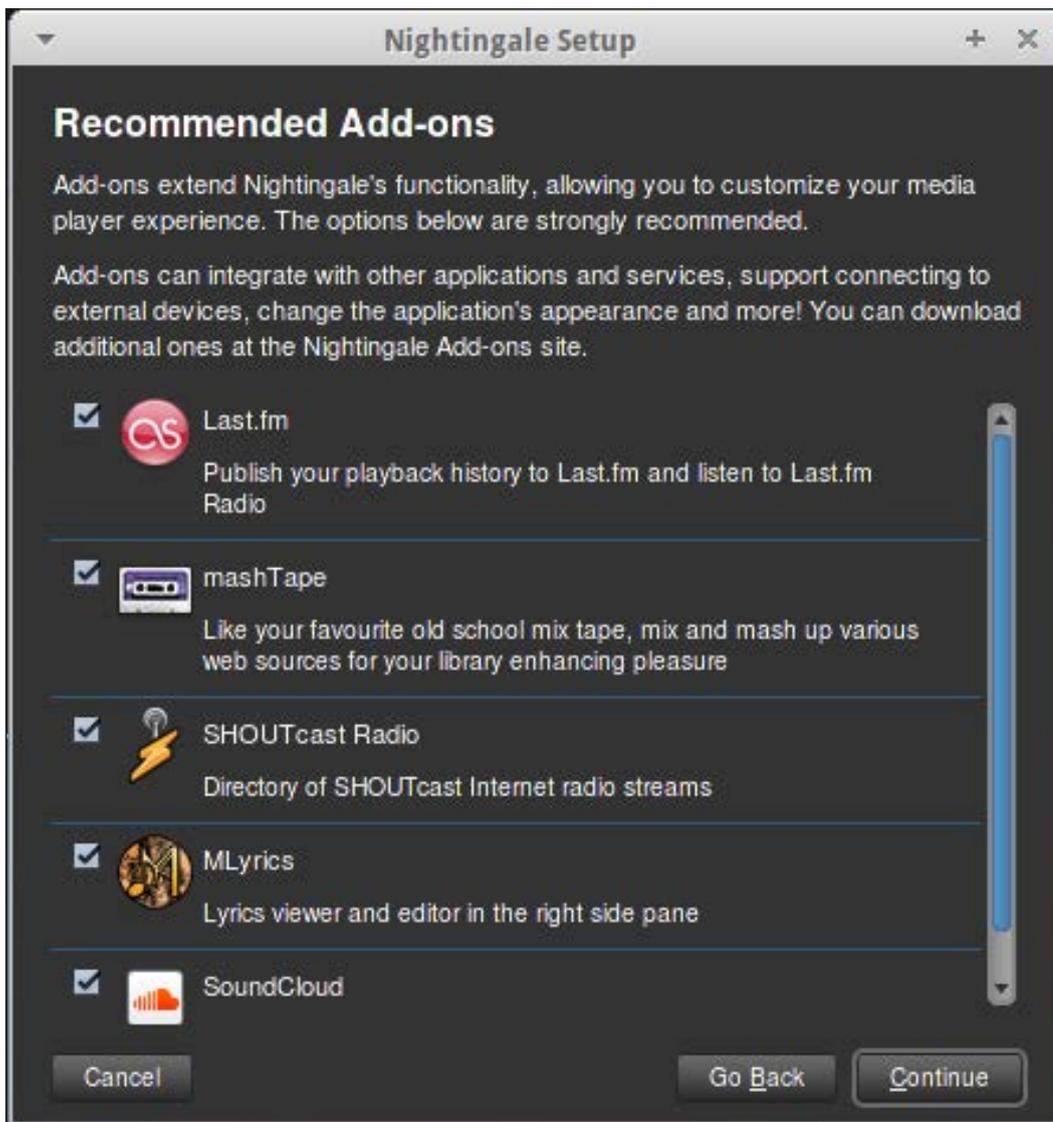


Figure 2. Plugins Recommended during Installation

educational experience. In Figure 1 you can see that as my Jonathan Coulton song plays, I automatically see the lyrics, plus instant information on the artist. If that sort of information doesn't interest you, no problem, Nightingale is highly customizable with plugins, and there are dozens and dozens available from its Web site (Figure 2

shows a handful of plugins recommended during the installation process).

Every music-playing software package I've tried has disappointed me in one way or another. In my brief relationship with Nightingale, I haven't found a single thing to dislike. The latest version even provides integration

into Ubuntu's Unity interface, if that's the desktop environment you prefer. Due to its simple interface, extendible underpinnings, and its continued devotion to the Linux desktop, Nightingale earns this month's Editors' Choice award. Get it for your computer today: <http://www.getnightingale.com>.

—**SHAWN POWERS**



REUVEN M.
LERNER

Compojure

In this article, Reuven shows how to connect a simple Clojure Web app to a PostgreSQL database.

In my last article, I started discussing Compojure, a Web framework written in the Clojure language. Clojure already has generated a great deal of excitement among software developers, in that it combines the beauty and expressive elegance of Lisp with the efficiency and ubiquity of the Java Virtual Machine (JVM). Clojure has other traits as well, including its famous use of software transactional memory (STM) to avoid problems in multithreaded environments.

As a Web developer and a longtime Lisp aficionado, I've been intrigued by the possibility of writing and deploying Web applications written in Clojure. Compojure would appear to be a simple framework for creating Web applications, built on lower-level systems, such as "ring", which handles HTTP requests.

In my last article, I explained how to create a simple Web application using the "lein" system, modify the project.clj configuration file and determine the HTML returned in response to

a particular URL pattern ("route"). Here, I try to advance the application somewhat, looking at the things that are typically of interest to Web developers. Even if you don't end up using Clojure or Compojure, I still think you'll learn something from understanding how these systems approach the problem.

Databases and Clojure

Because Clojure is built on the JVM, you can use the same objects in your Clojure program as you would in a Java program. In other words, if you want to connect to a PostgreSQL database, you do so with the same JDBC driver that Java applications do.

Installing the PostgreSQL JDBC driver requires two steps. First, you must download the driver, which is available at <http://jdbc.postgresql.org>. Second, you then must tell the JVM where it can find the classes that are defined by the driver. This is done by setting (or adding to) the CLASSPATH environment variable—that is, put the driver in:

```
export CLASSPATH=/home/reuven/Downloads:$CLASSPATH
```

Once you have done that, you can tell your Clojure project that you want to include the PostgreSQL JDBC driver by adding two elements to the `:dependencies` vector within the `defproject` macro:

```
(defproject cjtest "0.1.0-SNAPSHOT"
  :description "FIXME: write description"
  :url "http://example.com/FIXME"
  :dependencies [[org.clojure/clojure "1.5.1"]
                [compojure "1.1.5"]
                [hiccup "1.0.3"]
                [org.clojure/java.jdbc "0.2.3"]
                [postgresql "9.1-901.jdbc4"]]
  :plugins [[lein-ring "0.8.5"]]
  :ring {:handler cjtest.handler/app}
  :profiles
  {:dev {:dependencies [[ring-mock "0.1.5"]]]})
```

Now you just need to connect to the database, as well as interact with it. Assuming you have created a database named “cjtest” on your local PostgreSQL server, you can use the built-in Clojure REPL (`lein repl`) to talk to the database. First, you need to load the database driver and put it into an “sql” namespace that will allow you to work with the driver:

```
(require '[clojure.java.jdbc :as sql])
```

Then, you need to tell Clojure the host, database and port to which

you want to connect. You can do this most easily by creating a “db” map to build the query string that PostgreSQL needs:

```
(def db {:classname "org.postgresql.Driver"
        :subprotocol "postgresql"
        :subname (str "//" "localhost" ":" 5432 "/" "cjtest")
        :user "reuven"
        :password ""})
```

With this in place, you now can issue database commands. The easiest way to do so is to use the `with-connection` macro inside the “sql” namespace, which connects using the driver and then lets you issue a command. For example, if you want to create a new table containing a serial (that is, automatically updated primary key) column and a text column, you could do the following:

```
(sql/with-connection db
  (sql/create-table :foo [:id :serial] [:stuff :text]))
```

If you then check in `psql`, you’ll see that the table has indeed been created, using the types you specified. If you want to insert data, you can do so with the `sql/insert-values` function:

```
(sql/with-connection db (sql/insert-values
  ➔:foo [:stuff] ["first post"]))
```

Next, you get back the following map, indicating not only that the data was inserted, but also that it automatically was given an ID by PostgreSQL's sequence object:

```
{:stuff "first post", :id 1}
```

What if you want to retrieve all of the data you have inserted? You can use the `sql/with-query-results` function, iterating over the results with the standard `doseq` function:

```
(sql/with-connection db
  (sql/with-query-results resultset ["select * from foo"]
    (doseq [row resultset] (println row))))
```

Or, if you want only the contents of the "stuff" column, you can use:

```
(sql/with-connection db
  (sql/with-query-results resultset ["select * from foo"]
    (doseq [row resultset] (println (:stuff row)))))
```

Databases and Compojure

Now that you know how to do basic database operations from the Clojure REPL, you can put some of that code inside your Compojure application. For example, let's say you want to have an appointment calendar. For now, let's assume that there already is a PostgreSQL "appointments" database defined:

```
CREATE TABLE Appointments (
  id SERIAL,
  meeting_at TIMESTAMP,
  meeting_with TEXT,
  notes TEXT
);

INSERT INTO Appointments (meeting_at, meeting_with, notes)
VALUES ('2013-july-1 12:00', 'Mom', 'Always good to see Mom');
```

You'll now want to be able to go to `/appointments` in your Web application and see the current list of appointments. To do this, you need to add a route to your Web application, such that it'll invoke a function that then goes to the database and retrieves all of those elements.

Before you can do so, you need to load the PostgreSQL JDBC driver into your Clojure application. You can do this most easily in the `:require` section of your namespace declaration in `handler.clj`:

```
(ns cjtest.handler
  (:use compojure.core)
  (:require [compojure.handler :as handler]
            [compojure.route :as route]
            [clojure.java.jdbc :as sql]))
```

(I did this manually in the REPL with the "require" function, with slightly different syntax.)

You then include your same

definition of “db” in handler.clj, such that your database connection string still will be available.

Then, you add a new line to your `defroutes` macro, adding a new `/appointments` URL, which will invoke the `list-appointments` function:

```
(defroutes app-routes
  (GET "/" [] "Hello World")
  (GET "/appointments" [] list-appointments)
  (GET "/fancy/:name" [name] say-hello)
  (route/resources "/")
  (route/not-found "Not Found"))
```

Finally, you define `list-appointments`, a function that executes an SQL query and then grabs the resulting records and turns them into a bulleted list in HTML:

```
(defn list-appointments
  [req]
  (html
    [:h1 "Current appointments"]
    [:ul
      (sql/with-connection db
        (sql/with-query-results rs ["select * from appointments"]
          (doall
            (map format-appointment rs))))]))
```

Remember that in a functional language like Clojure, the idea is to get the results from the database and then process them in some way,

handing them off to another function for display (or further processing). The above function produces HTML output, using the Hiccup HTML-generation system. Using Hiccup, you easily can create (as in the above function) an H1 headline, followed by a “ul” list.

The real magic happens in the call to `sql/with-query-results`. That function puts the results of your database call in the `rs` variable. You then can do a number of different things with that resultset. In this case, let’s turn each record into an “li” tag in the final HTML. The easiest way to do that is to apply a function to each element of the resultset. In Clojure (as in many functional languages), you do this with the `map` function, which transforms a collection of items into a new collection of equal length.

What does the `format-appointment` function do? As you can imagine, it turns an appointment record into HTML:

```
(defn format-appointment [one-appointment]
  (html [:li (:meeting_at one-appointment)
        " : "
        (:meeting_with one-appointment)
        " (" (:notes one-appointment) ")"] ]))
```

In other words, you’ll treat the

record as if it were a hash and then retrieve the elements (keys) from that hash using Clojure's shorthand syntax for doing so. You wrap that up into HTML, and then you can display it for the user. The advantage of decomposing your display functionality into two functions is that you now can change the way in which appointments are displayed, without modifying the main function that's called when `/appointments` is requested by the user.

Inserting Data

Let's say you also want to insert data into your appointment book. To do that, you need an HTML form that then submits itself to a URL on your site. Let's first create a simple form—as always, written as a function:

```
(defn new-meeting-form
  [ req ]
  (html [:form {:method "POST" :action "/create-meeting"}
        [:p "Meeting at (in 2013-06-28T11:08 format): "
         ➤[:input {:type "text" :name "meeting_at"}]]
        [:p "Meeting with: " [:input {:type "text"
         ➤:name "meeting_with"}]]
        [:p "Notes: " [:input {:type "text" :name "notes"}]]
        [:p [:input {:type "submit" :value "Add meeting"}]]))
```

Notice how the Hiccup library

again lets you define HTML tags easily. In this case, because it's a form, you need to tell the form to which URL it should be submitted. So in this example, that'll be the `/create-meeting` URL. Thus, you need to define both `/new-meeting` and `/create-meeting` in your `defroutes` macro call:

```
(defroutes app-routes
  (GET "/" [] "Hello World")
  (GET "/meetings" [] list-meetings)
  (GET "/new-meeting" [] new-meeting-form)
  (POST "/create-meeting" [] create-meeting)
  (GET "/fancy/:name" [name] say-hello)
  (route/resources "/")
  (route/not-found "Not Found"))
```

As you can see, the routes distinguish between GET and POST requests. Thus, a GET request to `/create-meeting` will not have any effect (that is, it will result in the "not found" message being displayed); a POST request is needed to make it work.

Everything comes together when you want to add a new meeting to your database. You get the parameters from the submitted form and then insert them into the database.

I'm still learning about Clojure and Compojure and continue

Listing 1. handler.clj: Source Code for the Simple Appointment-Book System

```
(ns cjtest.handler
  (:use compojure.core hiccup.core clj-time.format clj-time.coerce)
  (:require [compojure.handler :as handler]
            [compojure.route :as route]
            [clojure.java.jdbc :as sql]))

(defn say-hello
  [req]
  (html [:p [:b "Hello, " (get (get req :route-params) :name) ]]))

(def db {:classname "org.postgresql.Driver"
        :subprotocol "postgresql"
        :subname (str "//" "localhost" ":" 5432 "/" "cjtest")
        :user "reuven"
        :password ""})

(defn format-meeting [one-meeting]
  (html [:li (:meeting_at one-meeting)
         " : "
         (:meeting_with one-meeting)
         " (" (:notes one-meeting) ")" ]))

(defn new-meeting-form
  [ req ]
  (html [:form {:method "POST" :action "/create-meeting"}
         [:p "Meeting at (in 2013-06-28T11:08 format): " [:input
         ↳{:type "text" :name "meeting_at"}]]
         [:p "Meeting with: " [:input {:type "text"
         ↳:name "meeting_with"}]]
         [:p "Notes: " [:input {:type "text" :name "notes"}]]
         [:p [:input {:type "submit" :value "Add meeting"}]]]))

(defn list-meetings
  [req]
  (html
   [:h1 "Current meetings"]
   [:ul
    (sql/with-connection db
     (sql/with-query-results rs ["select * from appointments"]
      (doall
       (map format-meeting rs))))]))

(defn create-meeting
  [req]
  (sql/with-connection db
   (let [form-params (:form-params req)
         meeting-at-string (get form-params "meeting_at")
         meeting-at-parsed (clj-time.format/parse
                             (clj-time.format/formatters
                              :date-hour-minute)
                              meeting-at-string)
         meeting-at-timestamp (clj-time.coerce/to-timestamp
                               ↳meeting-at-parsed)
         meeting-with (get form-params "meeting_with")
         notes (get form-params "notes")]
     (sql/insert-values :appointments
                       [:meeting_at :meeting_with :notes]
                       [meeting-at-timestamp meeting-with notes]))
   "Added!"))

(defroutes app-routes
  (GET "/" [] "Hello World")
  (GET "/meetings" [] list-meetings)
  (GET "/new-meeting" [] new-meeting-form)
  (POST "/create-meeting" [] create-meeting)
  (GET "/fancy/:name" [name] say-hello)
  (route/resources "/")
  (route/not-found "Not Found"))

(def app
  (handler/site app-routes))
```

to discover new libraries of functions that can make it easier to create HTML forms and work with databases. For example, I've recently discovered SQLKorma, a library that seems almost like Ruby's ActiveRecord, in that it provides a DSL that creates database queries.

The power of Clojure, like all Lisps, is partly based on the idea that you do everything in small steps and then combine those steps for the full power. Here, for example, is the function I wrote to add a new record (meeting) to the database:

```
(defn create-meeting
  [req]
  (sql/with-connection db
    (let [form-params (:form-params req)
          meeting-at-string (get form-params "meeting_at")
          meeting-at-parsed (clj-time.format/parse
                             (clj-time.format/formatters
                              :date-hour-minute)
                              meeting-at-string)
          meeting-at-timestamp (clj-time.coerce/to-timestamp
                                meeting-at-parsed)
          meeting-with (get form-params "meeting_with")
          notes (get form-params "notes")]
      (sql/insert-values :appointments
                        [:meeting_at :meeting_with :notes]
                        [meeting-at-timestamp meeting-with notes]))
    "Added!"))
```

The first and final parts of the function are similar in many ways to the database row insertion that you executed outside Compojure. You use `sql/with-connection` to connect to a database, and within that use `sql/insert-values` to insert a row into a specific table.

The interesting part of this function is, I believe, what happens in the middle. Using the "let" form, which performs local bindings of names to values, I can grab the values from the submitted HTML form elements, preparing them for entry into the database.

I further take advantage of the fact that Clojure's "let" allows you to bind names based on previously bound names. Thus, I can set `meeting-at-string` to the HTML form value, and then `meeting-at-parsed` to the value I get after converting the string to a parsed Clojure value, and then `meeting-at-timestamp` to turn it into a data type that both Clojure and PostgreSQL can handle easily.

Much of the heavy lifting here is being done by the `clj-time` package, which handles a wide variety of different date/time packages.

In the end, you're able to go to `/new-meeting`, enter appropriate data into the HTML form and save



DAVE TAYLOR

Web Administration Scripts

After some unpleasant experiences of his own, Dave explains how to create a script to detect DDOS attacks on a Web server.

Phew. I'm done with that *Cribbage* game coding, after months of shell script programming in directions doubtless unanticipated by the original Bash authors. It mostly worked, but after publishing last month's column, I did realize there are a few niggling bugs in the scoring code. Those, however, are now an exercise for you, dear reader, to identify and fix. Because you need homework, right?

During the past month or so, I've also been dealing with an aggressive DDOS (that's a "distributed denial of service") attack on my server, one that's been a huge pain, as you might expect. What's odd is that with multiple domains on the same server, it's one of my less-popular sites that seems to have been the target of the attacks.

So, that's the jumping off point for this article's scripts: analyzing log files to understand what's going on and why.

To start, a handy check is to see how many processes are running, because my DDOS was characterized by a ridiculous number of comment and search scripts being triggered—hundreds a minute. How to check?

The `ps` command offers a list of running processes at any given time, but for many versions, all you see is the Web server "httpd" without any further details. The `-C` cmd flag narrows down output only to those processes, like this:

```
: ps -C httpd
  PID TTY          TIME CMD
20225 ?                00:13:21 httpd
```

```
28162 ?      00:00:01 httpd      91
...
5681 ?      00:00:00 httpd      121
5683 ?      00:00:00 httpd <defunct> 120
116
```

(Note the “defunct” process that’s about to vanish.)

So one easy test is to see how many httpd processes are running:

```
$ ps -C httpd | wc -l
108
```

That seems like a lot, but this server is hosting several sites, including the super-busy AskDaveTaylor.com tech-support site, which sees more than 100k hits/day. So how does this vary over time? Hmm...still working on the command line:

```
$ while /bin/true
> do
>   ps -C httpd | wc -l
>   sleep 5
> done
108
107
103
99
94
91
87
84
```

So there’s a max of 121 and a min of 87. But, what if I actually want to analyze this and get min, max and average over a longer period of time? Here’s how I solve it:

```
#!/bin/sh
# Calculates the number of processes running that matches
# a set pattern over time, producing min, max and average.
min=999; max=0; average=0; tally=0; sumtotal=0
pattern="httpd" # ps -C pattern
while /bin/true
do
  count=$(ps -C $pattern | wc -l)
  tally=$(( $tally + 1 ))
  if [ $count -gt $max ] ; then
    max=$count
  fi
  if [ $count -lt $min ] ; then
    min=$count
  fi
  sumtotal=$(( $sumtotal + $count ))
  average=$(( $sumtotal / $tally ))
  echo "Current ps count=$count: min=$min, max=$max,
tally=$tally
  and average=$average"
  sleep 5 # seconds
done
exit 0
```

Notice in the script that I'm not falling into the trap of calculating the average by having a running average and somehow factoring in the latest value as a diminishing additive, but instead I use a `sumtotal` variable that keeps having the latest processor count added. That divided by `tally` is always the average, although at some point this probably would be greater than `MAXINT (2**32)` and would start to produce bad results. On a modern computer, however, that should take a while. (And the quantum, the period of time between iterations, also can be adjusted. Five seconds might be too granular for a process that's going to be run for hours or even days.)

The following are the first few lines of output. Notice how the `min` and `max` vary as the different values are calculated:

```
sh processes.sh
Current ps count=132: min=132, max=132, tally=1 and average=132
Current ps count=128: min=128, max=132, tally=2 and average=130
Current ps count=124: min=124, max=132, tally=3 and average=128
Current ps count=123: min=123, max=132, tally=4 and average=126
```

If I let the script run for a longer period of time, the values become a bit more varied:

```
Current ps count=90: min=76, max=150, tally=70 and average=107
```

During the 15 minutes or so that I ran the script, an average of 107 "httpd" processes were running, with a minimum of 76 and a max of 150.

Armed with that information, another script could keep an eye on things via a cron job, like this:

```
#!/bin/sh
# DDOS - keep an eye on process count to
# detect a blossoming DDOS attack
pattern="httpd"
max=200 # avoid false positives
admin="ditaylor@gmail.com"
count="$(ps -C $pattern | wc -l)"
if [ $count -gt $max ] ; then
    echo "Warning: DDOS in process? Current httpd count =
    ➔$count" | sendmail $admin
fi
exit 0
```

That's a superficial solution, however, and it has two problems: 1) what I'd really like is to be able to identify the *potential* DDOS based on processor count and watch to see if it's sustained over the next few invocations of the script, and 2) once it's triggered, if it is a DDOS, in addition to everything else, I'll also start drowning in e-mail from this script saying essentially the same thing each time. Not good.

What the script needs is contextual memory so it can

differentiate between a sudden spike in traffic and a persistent DDOS attack. In the former case, the script might trigger positive, then the next time it runs, it's all within acceptable limits again. In the latter case, once the attack starts, it'll probably just accelerate.

That's the opposite of the e-mail non-repeat condition though, because in the latter case, I want to know that the e-mail has been sent and not send it again within, say, a 60-minute window.

I'll dig in to both of those

situations next month. For now, I need to get back to my server and keep bringing things back on-line, program by program, to try to avoid any problems. Stay tuned! ■

Dave Taylor has been hacking shell scripts for more than 30 years. Really. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at <http://www.DaveTaylorOnline.com>.

Send comments or feedback via <http://www.linuxjournal.com/contact> or to ljeditor@linuxjournal.com.

Powerful: Rhino



Rhino M4700/M6700

- Dell Precision M4700/M6700 w/ Core i7 Quad (8 core)
- 15.6"-17.3" FHD LED w/ X@1920x1080
- NVidia Quadro K5000M
- 750 GB - 1 TB hard drive
- Up to 32 GB RAM (1866 MHz)
- DVD±RW or Blu-ray
- 802.11a/b/g/n
- Starts at \$1190
- E6230, E6330, E6430, E6530 also available

- High performance NVidia 3-D on an FHD RGB/LED
- High performance Core i7 Quad CPUs, 32 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 X230/X230 Tablet

- ThinkPad X230/X230 tablet by Lenovo
- 12.1" HD LED w/ X@1366x768
- 2.6-2.9 GHz Core i7
- Up to 16 GB RAM
- 750 GB hard drive / 180 GB SSD
- Pen/finger input to screen, rotation
- Starts at \$2050
- T430, T530, W530 also available

Rugged: Tarantula



Tarantula CF-31

- Panasonic Toughbook CF-31
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 13.1" XGA TouchScreen
- 2.4-2.53 GHz Core i5
- Up to 8 GB RAM
- 320-750 GB hard drive / 512 GB SSD
- CF-19, CF-52, CF-H2 also available

EmperorLinux
...where Linux & laptops converge

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





KYLE RANKIN

DNSSEC

Part II: the Implementation

Now that you know the fundamentals behind DNSSEC, it's time for the implementation.

This article is the second in a series on DNSSEC. In the first one, I gave a general overview of DNSSEC concepts to lay the foundation for this article, which discusses how to enable DNSSEC for a zone using BIND. If you want to deploy DNSSEC but aren't sure what I mean when I say KSK, ZSK, DLV or DS record, you may want to go back to Part I to refresh yourself on the concepts, because in this article, I'm going to dive right in to implementation.

Adding DNSSEC to a zone using BIND involves a few extra steps on top of what you normally would do to configure BIND as a master for your zone. First, you will need to generate a Key-Signing Key (KSK) and Zone-Signing Key (ZSK), then update the zone's config and sign it with the keys. Finally,

you will reconfigure BIND itself to support DNSSEC. After that, your zone should be ready, so if your registrar supports DNSSEC, you can update it or otherwise use DLV with a provider like dlv.isc.org. Now, let's look at the steps in more detail using my greenfly.org zone as an example.

Make the Keys

The first step is to generate the KSK and ZSK for your zone. As I mentioned in my previous article, the KSK is used only to sign ZSKs in the zone and to provide a signature for the zone's parent to sign, while ZSKs sign the records in each zone. Having separate keys also allows you to create a stronger KSK and have a weaker ZSK that you can rotate out each month. So first,

Adding DNSSEC to a zone using BIND involves a few extra steps on top of what you normally would do to configure BIND as a master for your zone.

let's create a KSK for greenfly.org using `dnssec-keygen`:

```
$ cd /etc/bind/  
$ dnssec-keygen -a RSASHA1 -b 2048 -n ZONE -f KSK greenfly.org
```

By default, the `dnssec-keygen` command dumps the generated keys in the current directory, so change to the directory in which you store your BIND configuration. The `-a` and `-b` arguments set the algorithm (RSASHA1) and key size (2048 bit), while the `-n` option tells `dnssec-keygen` what kind of key it is creating (a ZONE key). You also can use `dnssec-keygen` to generate keys for DDNS and other BIND features, so you need to be sure to specify this is for a zone. I also added a `-f KSK` option that tells `dnssec-keygen` to set a bit that denotes this key as a KSK instead of a ZSK. Finally, I specified the name of the zone this key is for: `greenfly.org`. This command should create two files: a `.key` file, which is the public key published in the zone, and a `.private` file, which is the private key and should be treated like a

secret. These files start with a K, then the name of the zone, and then a series of numbers (the latter of which is randomly generated), so in my case, it created two files: `Kgreenfly.org.+005+10849.key` and `Kgreenfly.org.+005+10849.private`.

Next I need to create the ZSK. The command is very similar to the command to create the KSK, except I lower the bit size to 1024 bits, and I remove the `-f KSK` argument:

```
$ dnssec-keygen -a RSASHA1 -b 1024 -n ZONE greenfly.org
```

This command creates two other key files: `Kgreenfly.org.+005+58317.key` and `Kgreenfly.org.+005+58317.private`. Now I'm ready to update and sign my zone.

Update the Zone File

Now that each key is created, I need to update my zone file for `greenfly.org` (the file that contains my SOA, NS, A and other records) to include the public KSK and ZSK. In BIND, you can achieve this by adding `$INCLUDE` lines to the end of your zone. In my case, I added

these two lines:

```
$INCLUDE Kgreenfly.org.+005+10849.key ; KSK
$INCLUDE Kgreenfly.org.+005+58317.key ; ZSK
```

Sign the Zone

Once the keys are included in the zone file, you are ready to sign the zone itself. You will use the `dnssec-signzone` command to do this:

```
$ dnssec-signzone -o greenfly.org -k Kgreenfly.org.+005+10849 \
  db.greenfly.org Kgreenfly.org.+005+58317.key
```

In this example, the `-o` option specifies the zone origin, essentially the actual name of the zone to update (in my case, `greenfly.org`). The `-k` option is used to point to the name of the KSK to use to sign the zone. The last two arguments are the zone file itself (`db.greenfly.org`) and the name of the ZSK file to use.

If you are using DLV, you will add an extra `-l` option to specify the DLV server you are using:

```
$ dnssec-signzone -l dlv.isc.org -o greenfly.org -k \
  Kgreenfly.org.+005+10849 db.greenfly.org \
  Kgreenfly.org.+005+58317.key
```

In either case, the command will create a new `.signed` zone file (in my case, `db.greenfly.org.signed`)

that contains all of your zone information along with a lot of new DNSSEC-related records that list signatures for each RRSET in your zone. If you aren't using DLV, it also will create a `dsset-zonename` file that contains a DS record you will use to get your zone signed by the zone parent. If you are using DLV, you will get a `dlvset-zonename` file. Any time you make a change to the zone, simply update your regular zone file like you normally would, then run the `dnssec-signzone` command to create an updated `.signed` file. Some administrators recommend even putting the `dnssec-signzone` command in a cron job to run daily or weekly, as by default the key signatures will expire after a month if you don't run `dnssec-signzone` in that time.

Reconfigure Zone's BIND Config

Now that you have a new `.signed` zone file, you will need to update your zone's config in BIND so that it uses it instead of the plain-text file, which is pretty straightforward:

```
zone "greenfly.org" {
    type master;
    file "/etc/bind/db.greenfly.org.signed";
    allow-transfer { slaves; };
};
```

Enable DNSSEC Support in BIND

Next, update the options that are enabled in your main BIND configuration file (often found in `named.conf` or `named.conf.options`), so that DNSSEC is enabled, the server attempts to validate DNSSEC for any recursive queries and DLV (DNSSEC Lookaside Validation) is supported:

```
options {
    dnssec-enable yes;
    dnssec-validation yes;
    dnssec-lookaside auto;
};
```

When you set `dnssec-lookaside` to `auto`, BIND automatically will trust the DLV signature it has for `dlv.isc.org` as it's included with the BIND software. Alternatively, you can add a DLV key manually if you add an additional BIND option and trusted key:

```
options { dnssec-lookaside . trust-anchor dlv.isc.org. };
trusted-keys {
    dlv.isc.org. 257 3 5
    "BEEAAAPHMu/5onzrEE7z1egmhg/WP00+juozrW3euWEn4MxDCE1+1Ly2
    brhQv5rN32RktMzX6Mj70jdzeND4XknW58dnJNPCxn8+jAG12FZLK8t+
    1uq4W+nnA3q02+DL+k6BD4mewMLbIYFwe0PG73Te9fZ2kJb56dhgMde5
    ymX4BI/oQ+cAK50/xvJv00FrF8kw6ucMTwF1gPe+jnGxPPEmHate/URK
    Y6Z2fkLoBAADLHQ9IrS2tryAe7mbBZVc0wIeU/Rw/mRx/vwMCTgNboM
    QKtUdvNXDrYD5HZws3xiRXF1Rf+a19UmZfSav/4NWLKjHzpT59k/VSt TDN0YUuWrBNh";
};
```

Once you are done changing your BIND configuration files, reload or restart BIND, and your zone should be ready to reply to DNSSEC queries.

Test DNSSEC

To test DNSSEC support for a zone, just add the `+dnssec` argument to `dig`. Here's an example query against `www.greenfly.org`:

```
$ dig +dnssec www.greenfly.org

;<<<> DiG 9.8.1-P1 <<<> +dnssec www.greenfly.org

;; global options: +cmd

;; Got answer:

;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 13093

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 5

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags: do; udp: 4096

;; QUESTION SECTION:

www.greenfly.org.          IN      A

;; ANSWER SECTION:

www.greenfly.org.         900    IN      A          64.142.56.172
www.greenfly.org.         900    IN      RRSIG     A 5 3 900 20130523213855
20130423213855 58317 greenfly.org.
cZS1G2Jj3FN80UrU4W+LbpCJlvVa+3yos1ni5V0pct4x41WvXGQNoh1G
/uFFJ62YRYXskL/c17wiAEIqsJ00/wzek5KFWAoiJ3zW05119c/8KPGF
7LzmEumdAVM2MmrPVu+PKGf1PlfofjwJLbgVhyYqepbbD8xv3bmg0Np YnM=

;; AUTHORITY SECTION:

greenfly.org.             900    IN      NS         ns2.greenfly.org.
```


What is a **Superhero** without the right tools?



No, we're not referring to costumed crime fighters who come to the rescue of others in trouble. We're talking about IT operations personnel - people with complex skills and a daunting job, who fight downtime, performance slowdowns, and other evils to keep your apps running 24x7. They have to be ready to take action if there is trouble brewing in the system – and being ready involves having the right tools.

ManageEngine provides the right set of monitoring tools for your IT operations team, enabling them to keep track of the performance of their complex apps from both within and outside their data center.

www.manageengine.com/apm

www.site24x7.com

- ⊙ Application Performance Monitoring
- ⊙ End User Experience Monitoring
- ⊙ Anomaly Detection
- ⊙ Automated Dependency Mapping
- ⊙ Deep Transaction Monitoring
- ⊙ Integrated Management Console





SHAWN POWERS

Protect Your Ports with a Reverse Proxy

Serve all your Web applications through a single server—no more port numbers!

In my last article, I discussed Apache Tomcat, which is the ideal way to run Java applications from your server. I explained that you can run those apps from Tomcat's default 8080 port, or you can configure Tomcat to use port 80. But, what if you want to run a traditional Web server *and* host Java apps on port 80? The answer is to run a reverse proxy.

The only assumption I make here is that you have a Web-based application running on a port other than port 80. This can be a Tomcat app, like I discussed in my last article, or it can be any Web application that has its interface via the Web (such as Transmission, Sick Beard and so on). The other scenario I cover here is running a Web app from a second server, even if it's on port 80, but you want it to be accessed from your central Web server.

(This is particularly useful if you have only one static IP to use for hosting.)

The way reverse proxying works, at least with the Apache Web server, is that every application is configured as a virtual host. Just like you can host multiple Web sites from a single server using virtual hosting, you also can host separate Web apps as virtual hosts from that same server. It's not terribly difficult to configure, but it's very useful in practice. First things first. On your server, you have the Web server installed (Figure 1). You also have a Web application on port 8080 (Figure 2). Along with the working Apache Web server, you need to make sure virtual hosting (by name) is enabled.

Enabling Name-Based Virtual Hosts

Enabling name-based virtual hosting on Apache is extremely common, and

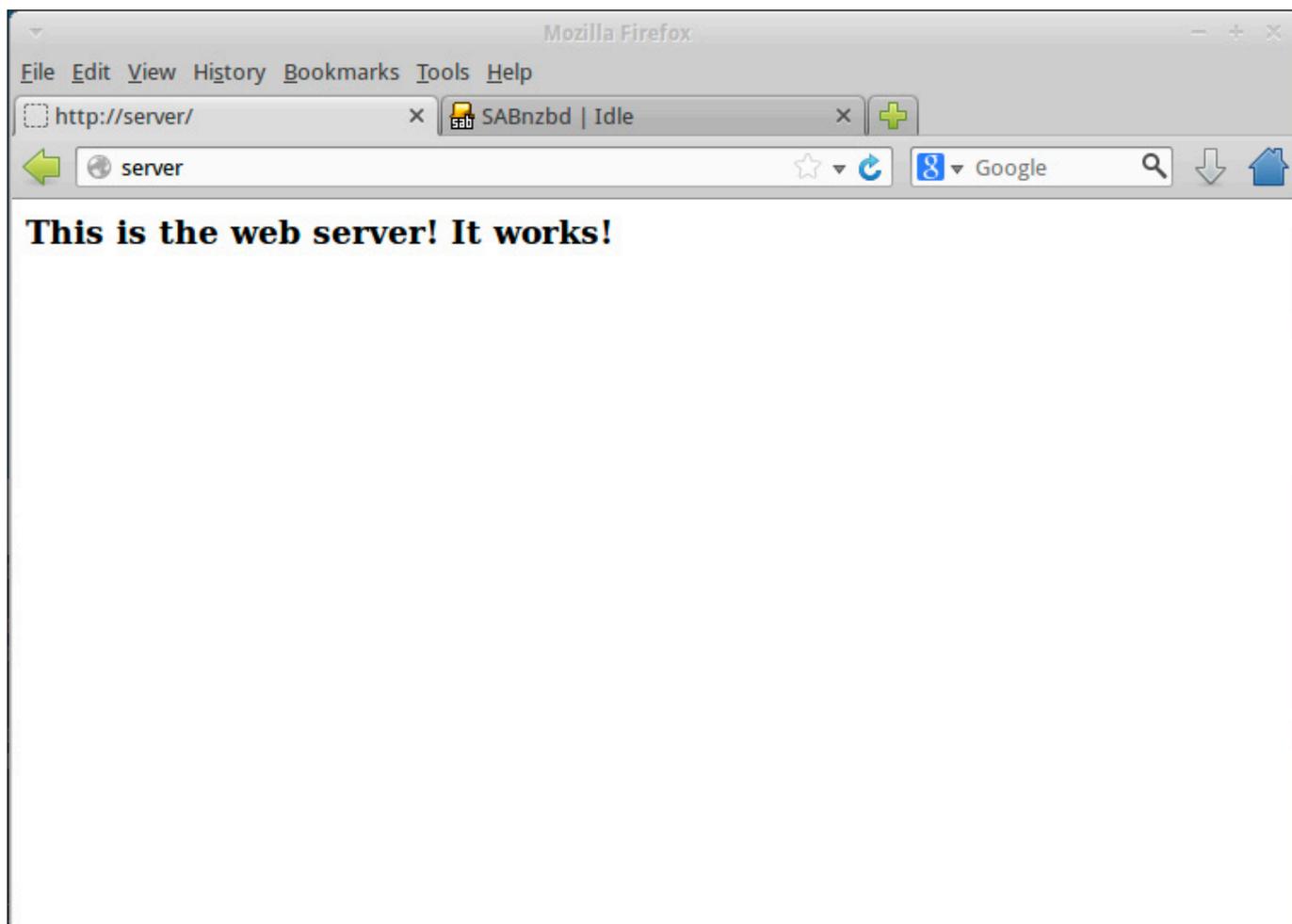


Figure 1. I have Apache installed, and it's hosting a very simple page. on port 80.

it's very simple to do. Depending on what distribution you're using, the "proper" location for enabling name-based virtual hosting may differ. The nice thing about Apache, however, is that generally as long as the directive is specified somewhere in the configurations, Apache will honor it.

My local test server is running Ubuntu. In order to determine where the "proper" place to enable name-based virtual hosting is, I simply went to the `/etc/apache2` directory and executed:

```
grep NameVirtualHost *
```

That command searches for the `NameVirtualHost` directive, and it returned this:

```
root@server:/etc/apache2# grep NameVirtualHost *
ports.conf:NameVirtualHost *:80
ports.conf:  # If you add NameVirtualHost *:443 here,
              # you will also have to change
```

Those results tell me that the `NameVirtualHost` directive is specified

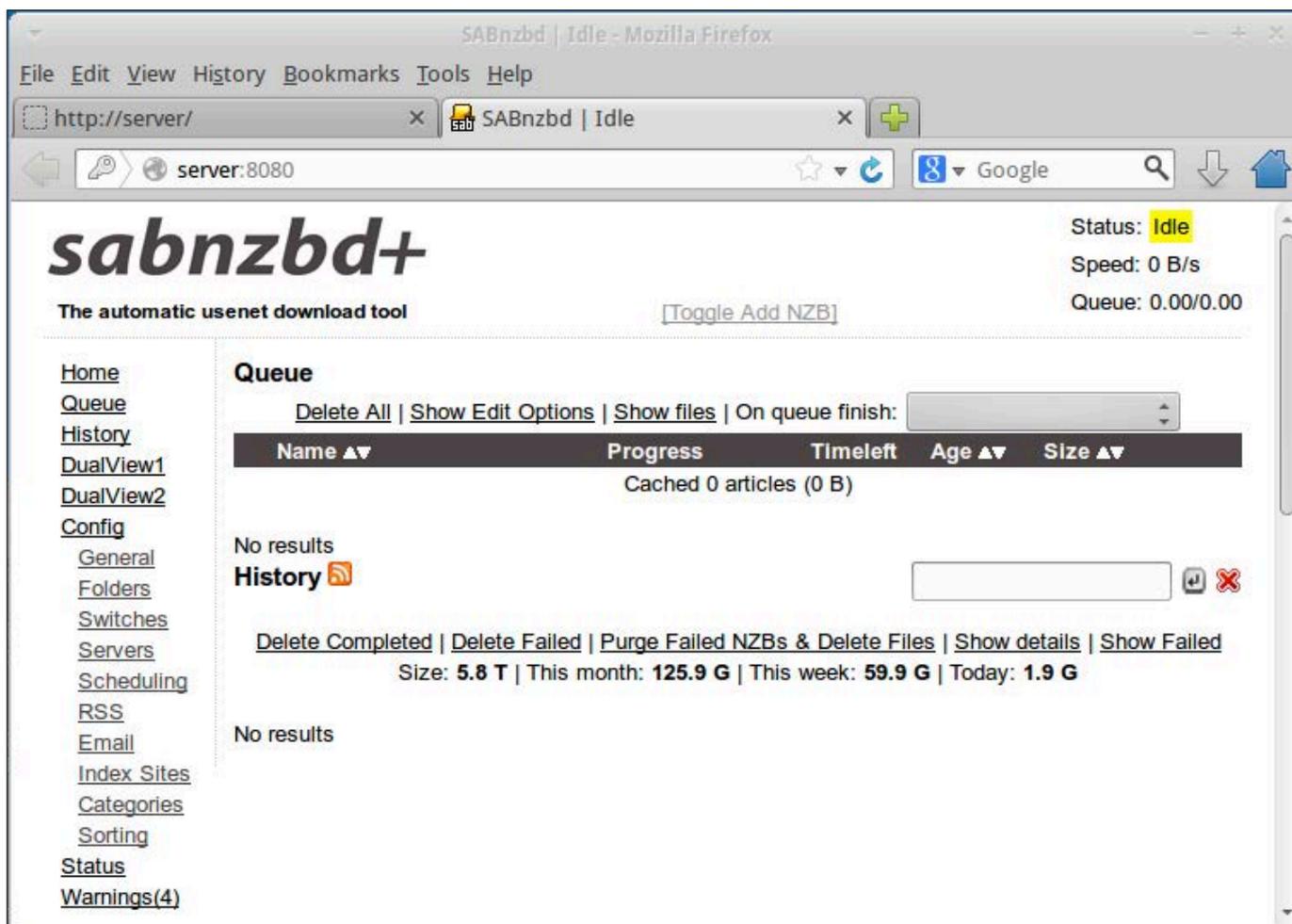


Figure 2. I have a Web application running on port 8080 on the server located at 192.168.1.11.

in the `/etc/apache2/ports.conf` file. (Note that `grep` will return only the lines that contain the search term, which is why it shows those two out-of-context lines above. The important thing is the filename `ports.conf`, which is what I was looking for.) Again, with Apache, it generally doesn't matter where you specify directives, but I like to stick with the standards of the particular distribution I'm using, if only for the sake of future administrators.

To enable name-based virtual

hosting, you simply uncomment:

```
NameVirtualHost *:80
```

from the file, and save it. If you can't find a file that contains such a directive commented out, just add the line to your `apache.conf` or `httpd.conf` file. Then you need to specify a `VirtualHost` directive for the virtual host you want to create. This process is the same whether you're making a traditional virtual host or a reverse proxy virtual host.

Creating a Virtual Host

As in the previous section of this article, it's important to note that the Apache configuration file layout will vary with distributions. In Ubuntu, there are two folders: sites-available and sites-enabled. The first has text files with snippets of code defining the individual virtual hosts, and the second has symbolic links to the files located in the sites-available folder. This seems complicated to be sure, but it's actually for convenience sake. You can define as many virtual hosts as you want in the sites-available folder, but until they're symbolically linked into the sites-enabled folder, they're not parsed by Apache.

Let's create a virtual host, but instead of making a traditional virtual host that defines a directory to look for files, let's define reverse proxy rules. Here is the file I created in sites-available (I explain each line next):

```
root@server:/etc/apache2# cat sites-available/reverseprox
<VirtualHost *:80>
    LoadModule proxy_module modules/mod_proxy.so
    LoadModule proxy_http_module modules/mod_proxy_http.so
    ServerName sab.mydomain.com
    ServerAlias sab
    ProxyRequests Off
    ProxyPass / http://192.168.1.11:8080/
    ProxyPassReverse / http://192.168.1.11:8080/
</VirtualHost>
```

First off, if it's not clear, the name

of the file I created is "reverseprox", and I created it in the /etc/apache2/sites-available folder. If you are using a different distribution, you may not have this sort of folder setup. You actually can add the VirtualHost directives directly to the apache.conf or httpd.conf file. Ubuntu just uses the folder structure for clarity and convenience.

Here's the line-by-line breakdown:

- `<VirtualHost *:80>` — this opens the stanza, and it means "listen on all IP addresses on port 80 for anyone requesting my server name".
- `LoadModule proxy_module modules/mod_proxy.so` and `LoadModule proxy_http_module modules/mod_proxy_http.so` — these lines load two separate modules. Note that although the module names look similar, they actually are two modules: `mod_proxy` and `mod_proxy_http`. Sometimes modules are loaded globally in another configuration file. That's okay to do, but this is just a way to make sure the required modules are loaded for your virtual host. (Note: if you get an error about "file not found" during startup, you might need to make a symbolic link to your system's modules folder. On my Ubuntu system that means

```
sudo ln -s /usr/lib/apache2/  
modules etc/apache2/.)
```

- `ServerName sab.mydomain.com` — this is the domain name the virtual host should listen for. If a request comes into Apache for “sab.mydomain.com”, it knows to use this virtual host declaration to respond. Of course, “sab.mydomain.com” is a generic example; you should use your actual domain name.
- `ServerAlias sab` — it’s possible to have multiple `ServerAlias` statements, but in this case, there’s only one. I’ve added “sab” all by itself as an alias for Apache to listen for. It will use a request for “sab” the same way it uses a request for “sab.mydomain.com”—this is simply an alias.
- `ProxyRequests Off` — this is actually the default setting for the `ProxyRequests` directive. I always add it to my `VirtualHost` stanza anyway to make sure I’m not inadvertently allowing someone to use my server as an anonymous proxy. `ProxyRequests On` would allow others with access to your server to use it as a proxy, effectively hiding themselves from the Internet and making you responsible for their surfing! Hopefully, it’s clear why I specify “Off”, even though it’s the default setting.
- `ProxyPass / http://192.168.1.11:8080/` — this tells Apache that when someone requests the root-level folder of this virtual host to “serve” them the address listed. From end users’ perspectives, the alternate port, and possibly the alternate server address, will be hidden. They’ll see only the URL they entered to get to the virtual host. You can have multiple `ProxyPass` directives if you want a specific subfolder to be directed elsewhere. Apache is very flexible with what you can specify in a reverse proxy situation.
- `ProxyPassReverse / http://192.168.1.11:8080/` — this rule is what makes the reverse proxy work. It rewrites the response from the proxied server so that end users never see any information apart from the virtual hostname they’ve surfed to. Any responses from the underlying server (in this case, the server listening on port 8080) are rewritten on the fly so that it appears that the responses are coming directly from the virtual host server.
- `</VirtualHost>` — this closes the stanza, or the section defining the virtual host. In Ubuntu, this is a single file in the sites-available folder. It also could just be something tacked onto

the end of the `apache.conf` file in another distribution.

Making It All Work

Once you've created the virtual host declaration for the reverse proxy site, you need to reload Apache. Remember, if you're using Ubuntu, you need to create a symbolic link so that Apache reads your configuration from the `sites-enabled` folder. To do that, go into the `sites-enabled` folder, and type:

```
ln -s ../sites-available/reverseprox .
```

This will create a symbolic link from the `reverseprox` file you created to the `sites-available` folder. If you're using another distribution and just tacked that stanza to the end of the `apache.conf` file, you don't need to make any symbolic links.

Next, reload Apache. I actually prefer to restart Apache to make sure it loads up everything correctly, but a reload should do the trick. In Ubuntu, I do this:

```
sudo service apache2 restart
```

And, the reverse proxy should be ready to go. You just need to make sure your DNS points correctly to the server. The quickest way to do that, and make sure stuff is working, is to add a simple line to your workstation's

`/etc/hosts` file. I added this:

```
192.168.1.11    sab sab.mydomain.com
```

And, then I saved it. Next, I opened a browser, and surfed to "sab" instead of `192.168.1.11:8080`, and Figure 3 shows the results. Success!

Now What?

The great thing about using Apache's reverse proxy technique is that you're not limited to redirecting only to the same server on a different port. You can make a reverse proxy so that `google.yourdomain.com` returns the actual Google search engine. You'll just create a virtual host for `google.yourdomain.com`, and set the `ProxyPass` and `ProxyPassReverse` directives to point to `http://www.google.com/`. It's truly simple. In fact, a reverse proxy on your local network might be a way to provide access to an otherwise blocked Web site for your users. What if your Web-filtering policies blocked a particular news site, but your server had access? You could create a reverse proxy on your server that your users could connect to and get to the site without being filtered by your Web filter! (Another word of caution: this is why it's important to set `ProxyRequests` to `Off`, so they don't use your reverse proxy to circumvent *all* Web filtering!)

Dedicated Infrastructure for your Business



Dedicated Cloud

VMware vSphere® & vCloud®
Immediate Accessibility
Scalable Infrastructure

Dedicated Servers

Unlimited Traffic & Guaranteed Bandwidth
24/7 North American Support
Total Control

From:

\$527⁰⁰
/month

your Virtual Data Center

1 Month Free Trial

Call us: 1-855-684-5463 (toll free)

From:

\$39⁰⁰
/month

your Dedicated Server

Find our Entire Range

www.ovh.com/dedicated-servers



Dedicated Cloud



Dedicated Servers

For more details:



OVH.COM

Dedicated Infrastructure for your Business

or contact us: 1-855-684-5463 (toll free)

MEET US AT
vmworld 2013
BOOTH 1713

August 25-29
SAN FRANCISCO

vmware®
PARTNER NETWORK AWARD
2013 GLOBAL WINNER

vmware®
PARTNER NETWORK AWARD
2013 EMEA WINNER

LynuxWorks' LynxOS

The term "Internet of Things" is increasingly used to describe our present-day network of tens of billions of connected devices. In response to the increasing embedded nature of this network and the growing number and sophistication of threats to it, embedded software developer LynuxWorks presents LynxOS 7.0, the next generation of its popular RTOS. The core focus of version 7.0 of LynxOS is to give developers the tools to guard against threats at the operating system level, embedding military-grade security directly into its devices via features like access control lists, audit, quotas, local trusted path, account management, trusted menu manager and OpenPAM. This release also features networking support for common protocols utilized in both long- and short-haul networks. In order to satisfy demanding real-time QoS requirements of certain market segments, LynuxWorks has partnered with key middleware providers, such as Real-Time Innovations, Inc. (RTI), to port their offerings to the LynxOS platform.

<http://www.lnxw.com>



Real Time Logic's Mako Server

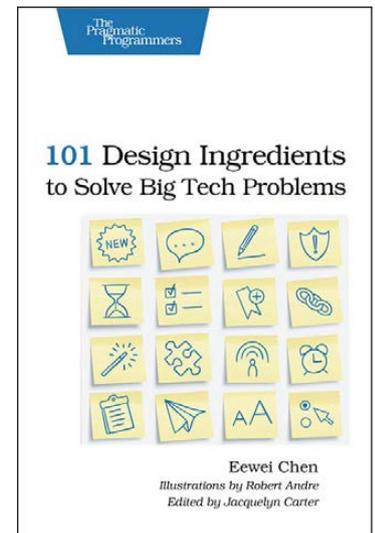
Real Time Logic boasts that its Mako Server Web application back end for Linux, Mac and Windows platforms can respond with 45,000 dynamic page requests in the same time that Apache outputs 25,000 less-compute-intensive static pages. Utilizing the easy-to-learn Lua scripting language, Mako Server offers fast, efficient development of Web applications, ranging from database-driven business applications to customized applications managing microcontroller-based devices, says Real Time Logic. In contrast to the typical approach requiring integration and configuration of components, such as Apache, PHP and SQL database, Mako Server brings all of these components together, bundling them into one unit so that the developer can immediately focus on application development for the desired platform or device. With the server, developers can bundle their applications into a single zip file, so users can download and run the application just as they would a Windows-based application.

<http://makoserver.net>

Ewei Chen's *101 Design Ingredients to Solve Big Tech Problems* (Pragmatic Bookshelf)

You might be a white-belt geek who is venturing into your first big technology project. Or, you could be a black-belt master geek who's been tackling big problems for years and needs a fresh approach to problem solving. Whatever your mastery level, the wisdom found on the pages of Ewei Chen's new book *101 Design Ingredients to Solve Big Tech Problems* may help you solve the daunting problems that vex you. Humorously illustrated, *101 Design Ingredients* is designed to help a technology team identify problems, share responsibilities and work better together. Part I features case studies that demonstrate how companies like Facebook and Dropbox blended ingredients from this book to solve specific business requirements for investment, innovation, leadership and more. Part II consists of the 101 problem-solving ingredients, grouped into project stages, to help one apply the right ingredient at the right time. The ingredients cover the spectrum a business needs to be successful.

<http://www.pragprog.com>



[PACKT] Alex Blewitt's *Eclipse Plugin Development by Example* (Packt Publishing)

A nice feature about Alex Blewitt's new book *Eclipse Plugin Development by Example: Beginner's Guide* is that one need not have prior experience in Eclipse plugin development or OSGi. With this book as a guide, Java developers who already are familiar with Eclipse as an IDE will embark on a full journey through plugin development, starting with an introduction to Eclipse plugins, continuing through packaging and culminating in automated testing and deployment. The included example code provides simple snippets that can be developed and extended to get users up and running quickly. A specific chapter on the differences between Eclipse 3.x and Eclipse 4.x presents a detailed view of the changes needed by applications and plugins when upgrading to the new model.

<http://www.packtpub.com>



redhat.

Red Hat Enterprise Virtualization 3.2

Sources at Red Hat call the new Red Hat Enterprise Virtualization 3.2 “a significant step forward for open-source virtualization”. A core element of Red Hat’s open hybrid cloud offerings, Red Hat Enterprise Virtualization is a mission-critical end-to-end, open-source virtualization infrastructure designed for enterprise users and global organizations. The platform is designed to meet an increasing industry need for open virtualization solutions without compromising performance, scalability, security or features. Version 3.2 adds support for Storage Live Migration; support for the latest industry-standard processors from Intel and AMD, including the Intel Haswell series and AMD Opteron G5 processors; and enhancements in storage management, networking management, fencing and power management, Spice console, logging and monitoring, and more. A new third-party plugin framework enables third parties to integrate new features and actions directly into the user interface; solutions from NetApp, Symantec and HP already are in development.

<http://www.redhat.com>

Epiq Solutions’ Matchstiq Z1

The most spot-on mantra for our current era is “do more with less”, and such is the accomplishment of Epiq Solutions’ Matchstiq Z1, a small form-factor software-defined radio (SDR) solution. Measuring only 2.2" x 4.6" x 0.9", the Matchstiq Z1 combines a Xilinx Zynq Z-7020 SOC running Linux with a flexible RF transceiver capable of tuning between 300MHz and 3.8GHz in a complete SDR solution. Epiq Solutions says that the Matchstiq Z1 provides a more capable signal processing system while maintaining the same footprint as the existing Matchstiq platform. The company further notes that users can combine a library of signal processing applications from Epiq Solutions or other signal processing frameworks, such as GNU Radio or REDHAWK, to enable countless capabilities to the Matchstiq Z1, including using it as an agile point-to-point data modem, LTE survey tool or portable spectrum analyzer. Development kits also are available for end users who want to create their own custom applications.



<http://www.epiqsolutions.com/matchstiq>

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.

**USING THE
R
ADVANCED
STATISTICAL
PACKAGE**

Are you ready for R?

MIHALIS TSOUKALOS

This article is about the R advanced statistical package. Despite its simple name, R is a wonderful piece of statistical software with many complex capabilities and an interpreted computer language—it's also free. Don't be afraid of R if you don't feel very comfortable with mathematics or statistics. This article presents some easy-to-understand and practical scenarios that illustrate the use of R.

R is a GNU project based on S, which is a statistics-specific language and environment developed at the famous AT&T Bell Labs. You can think of R as the free version of S. The R system distribution supports a large number of statistical procedures, including linear and generalized linear models, nonlinear regression models, time series analysis, classical parametric and nonparametric tests, clustering and smoothing. At the time of this writing, the current version of R is 3.0.1, which was released May 16, 2013.

You can use GUIs for R, and the most popular GUI, which also is my favorite, is called RStudio. However, I use only the command-line version of R for this article to keep things as general as possible.

Running R

Your Linux/UNIX distribution probably includes a ready-to-install

R package, so go ahead and install it. Alternatively, you can go to <http://cran.r-project.org> and download a precompiled binary or get the source code and compile it yourself. After installing it, typing R on your terminal will take you to the R shell. Once the R shell starts, you can start typing R commands. The initial R output on your screen should look similar to the following:

```
$ R

R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin12.3.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> q()
Save workspace image? [y/n/c]: n
mtsouk$
```

One of the first things you will want to learn is how to quit R. Typing `q()` quits the R shell and takes you back to the UNIX shell.

R keeps a history of all typed commands in a hidden file called `.Rhistory`. The `.Rhistory` file is stored inside the directory where you ran the R binary, so if you are running R from multiple directories, you will have multiple `.Rhistory` files on your computer.

The contents of a simple `.Rhistory` file look like this:

```
$ cat .Rhistory
install.packages("RCurl")
install.packages("RJSON")
install.packages("rjson")
install.packages("rgoogleanalytics")
install.packages("google")
source("./RGoogleAnalytics.R")
source("db.R")
summary(wwwwdatacomma)
q <- sqldf("SELECT count(*) FROM WWW", dbname = "WWW.sqlite")
q()
```

Notice that the `.Rhistory` file also includes erroneous commands that were typed but not executed, so don't trust everything you see in it.

In order to avoid retyping the same R code, you can create R scripts, which is a very handy R feature. A good practice is first

to try the commands one by one inside the R shell and then convert them into a script to save time. As always, don't forget to include comments in your code.

The `source()` command is used for calling an existing R script when you are inside the R shell. If you want to find help for the `source()` command (or any other existing command), simply type the following:

```
> ?source()
```

If you want to search for help, but you don't know the exact command, try the following:

```
> help.search("keywords to find")
```

R supports the use of the Tab key, as in the bash shell, so type the first letters of a command, press the Tab key, and R will help you find the rest of the command you are trying to type.

Installing an R Package

R has a large repository of existing packages, so you don't have to program everything from the beginning. There are two ways to install an R package:

1. Install a package that can be found

on CRAN (The Comprehensive R Archive Network) using the `install.packages()` function.

2. Download it to your computer and install it from the local file using the same `install.packages()` command but with different parameters.

The next section of this article shows examples of both installation methods.

The `library()` function, without any arguments, prints a list of all the installed packages. To get more-detailed output of all the installed R packages, you also can use the `installed.packages()` command. The `update.packages()` command will update the installed CRAN packages to their latest versions.

Communicating with Google Analytics

R can communicate with Google Analytics natively using an R package, so you can retrieve and perform statistical analysis of the Google Analytics data. The first step is to download the relevant R package from <https://code.google.com/p/r-google-analytics>, because CRAN does not contain the RGoogleAnalytics package. Make sure you don't download the ZIP file, because it is the Windows version of the R package.

The UNIX version is a `.tar.gz` file called `RGoogleAnalytics_1.3.tar.gz` (at the time of this writing).

Then, you need to install it manually using the following command, provided that the `RGoogleAnalytics_1.3.tar.gz` file is in your current working directory:

```
> install.packages("./RGoogleAnalytics_1.3.tar.gz",  
  ↳repos=NULL, type="source")
```

The first time I tried to install it, I got the following error messages:

```
> install.packages("./RGoogleAnalytics_1.3.tar.gz",  
  ↳repos=NULL, type="source")  
Warning in install.packages("./RGoogleAnalytics_1.3.tar.gz",  
  ↳repos = NULL, :  
  'lib = "/opt/local/Library/Frameworks/R.framework/Versions/  
  ↳3.0/Resources/library"' is not writable  
Would you like to use a personal library instead? (y/n) y  
Would you like to create a personal library  
~/Library/R/3.0/Library  
to install packages into? (y/n) y  
ERROR: dependencies 'rjson', 'RCurl' are not available for  
  ↳package 'RGoogleAnalytics'  
* removing '/Users/mtsouk/Library/R/3.0/library/  
RGoogleAnalytics'  
Warning message:  
In install.packages("./RGoogleAnalytics_1.3.tar.gz",  
  ↳repos = NULL, :  
  installation of package './RGoogleAnalytics_1.3.tar.gz'  
  ↳had non-zero exit status  
>
```

This error messages tells me I need to have the rjson and RCurl packages installed in advance. Both of them can be found on CRAN, and the following shows their installation process:

```
> install.packages('rjson')
Installing package into '/Users/mtsouk/Library/R/3.0/library'
(as 'lib' is unspecified)
...
...
The downloaded source packages are in
  '/private/var/folders/9m/8b9b4ttn6gvbwg7drb2jcp540000gn/
  ↳T/RtmpIBUmtw/downloaded_packages'
> install.packages('RCurl')
Installing package into '/Users/mtsouk/Library/R/3.0/library'
(as 'lib' is unspecified)
also installing the dependency 'bitops'

trying URL 'http://cran.cc.uoc.gr/src/contrib/bitops_1.0-5.tar.gz'
Content type 'application/x-gzip' length 8518 bytes
opened URL
=====
downloaded 8518 bytes

trying URL 'http://cran.cc.uoc.gr/src/contrib/RCurl_1.95-4.1.tar.gz'
Content type 'application/x-gzip' length 870915 bytes (850 Kb)
opened URL
=====
downloaded 850 Kb
...
...
** building package indices
** installing vignettes
** testing if installed package can be loaded
```

```
* DONE (RCurl)

The downloaded source packages are in
  '/private/var/folders/9m/8b9b4ttn6gvbwg7drb2jcp540000gn/
  ↳T/RtmpIBUmtw/downloaded_packages'
>
```

Finally, you can install the desired r-google-analytics R package without any problems:

```
> install.packages("./RGoogleAnalytics_1.3.tar.gz",
  ↳repos=NULL, type="source")
Installing package into '/Users/mtsouk/Library/R/3.0/library'
(as 'lib' is unspecified)
* installing *source* package 'RGoogleAnalytics' ...
** R
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (RGoogleAnalytics)
>
```

The contents of the RGoogleAnalytics directory are the following:

```
-rw-r--r--  1 mtsouk  staff   902 Jun  6 23:01 DESCRIPTION
-rw-r--r--  1 mtsouk  staff  2071 Jun  6 23:01 INDEX
drwxr-xr-x  7 mtsouk  staff   238 Jun  6 23:01 Meta
-rw-r--r--  1 mtsouk  staff    30 Jun  6 23:01 NAMESPACE
drwxr-xr-x  5 mtsouk  staff   170 Jun  6 23:01 R
drwxr-xr-x  7 mtsouk  staff   238 Jun  6 23:01 help
drwxr-xr-x  4 mtsouk  staff   136 Jun  6 23:01 html
```

To make sure that the RGoogleAnalytics package is installed properly, run the following command inside the R shell:

```
> library("RGoogleAnalytics")
Loading required package: rjson
Loading required package: RCurl
Loading required package: bitops
```

If your output is similar to the above, everything is fine, and you are ready to continue with the rest of the article. As you also can see in this output, if you try to load the RGoogleAnalytics package, it automatically will load the rjson, RCurl and bitops packages, so you don't need to load them manually inside your R scripts.

The RGoogleAnalytics package consists of the following two classes:

- R Google Analytics: this is the main R package class.
- Query Builder: this class simplifies the creation of queries.

The following is an R script (saved as a file called GA.R) that uses the Google Analytics R package (the line numbers were added to refer to the R code—those need not to be typed):

```
1 require("RGoogleAnalytics")
2 query <- QueryBuilder()
3 access_token <- query$authorize()
4 ga <- RGoogleAnalytics()
5 ga.profiles <- ga$GetProfileData(access_token)
6 # ga.profiles
7 query$Init(start.date = "2013-03-01",
8           end.date = "2013-04-01",
9           dimensions = "ga:date,ga:pagePath",
10          metrics = "ga:visits,ga:pageviews,ga:timeOnPage",
11          sort = "ga:visits",
12          #filters="",
13          #segment="",
14          max.results = 99,
15          table.id = paste("ga:",ga.profiles$id[1],
16          sep="",collapse=","),
17          access_token=access_token)
17 ga.data <- ga$GetReportData(query)
18 # head(ga.data)
```

Let me explain the R script line by line:

- 1: the first command loads the RGoogleAnalytics library and its dependencies.
- 2: defines a QueryBuilder variable that will be used when defining the query.
- 3: this command gets the required access token that will be generated by Google (Figure 1). You need to log in to Google Analytics using your favorite

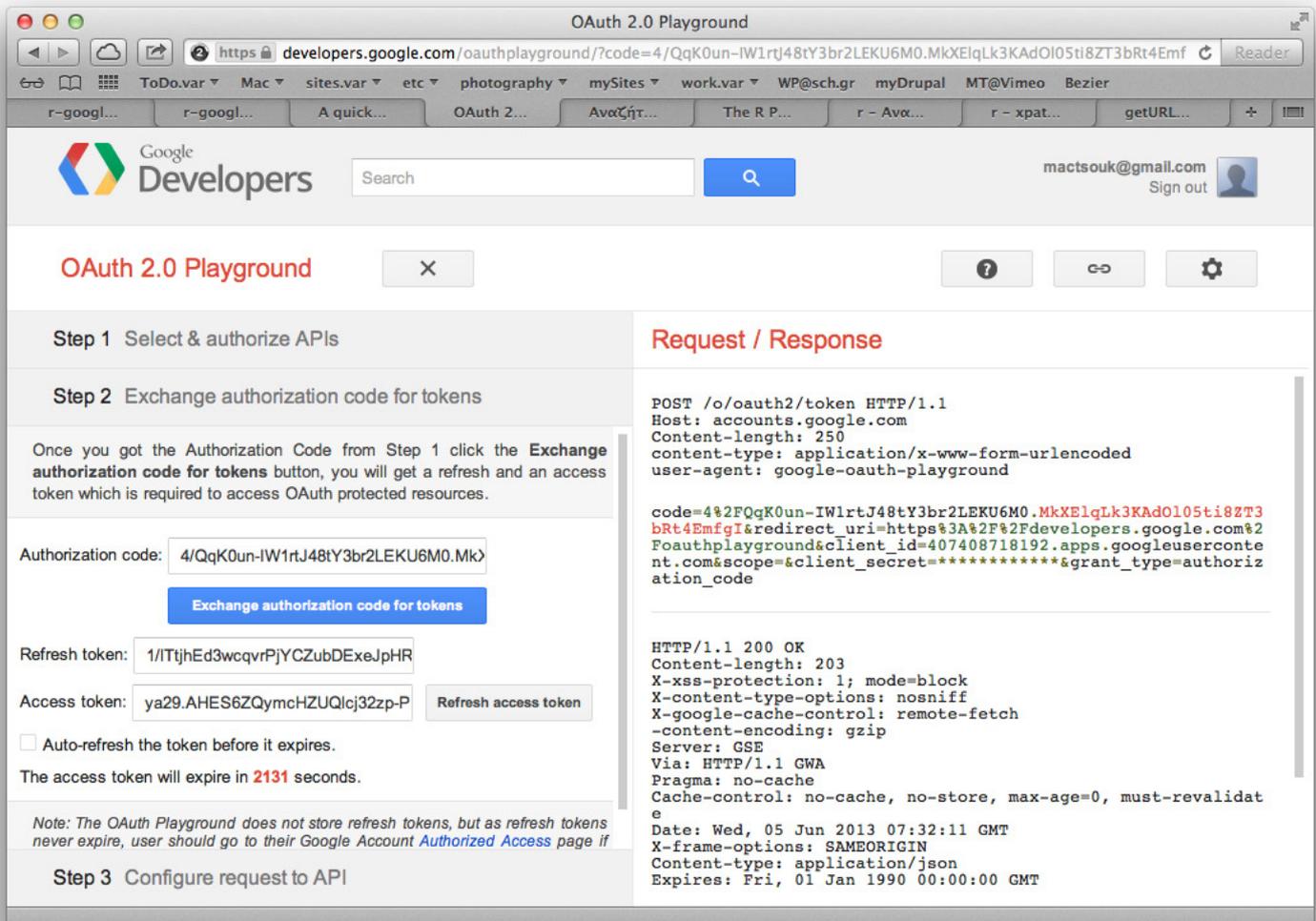


Figure 1. Input for the query \$authorize() Command

Web browser. As you also can see in Figure 1, for security reasons, the provided access token expires if you do not refresh it.

- 4: creates a new Google Analytics API object.
- 5: gets the available profiles that are connected to the Google Analytics account.

- 6: prints the available profiles. This is an optional step and is commented out.
- 7–16: defines the Query that will be used. There are many parameters; go to <https://developers.google.com/analytics/devguides> to learn more about them.
- 17: files a request to get the data from the API.

■ 18: allows you to look at the returned data. This is an optional step and is commented out (I think it's better to execute it manually).

In order to run the GA.R script, you can use the `source()` R command as follows (provided that GA.R is in your current working directory):

```
> source ("./GA.R")
Loading required package: RGoogleAnalytics
Loading required package: rjson
Loading required package: RCurl
Loading required package: bitops
```

The GA data extraction process requires an access token. To accept the access token from the OAuth 2.0 Playground, you need to follow certain steps in your browser. This access token will be valid only for one hour.

Here are the steps:

1) Authorize your Google Analytics account by providing your e-mail and password.

2) On the left side of the screen, click the button "Exchange authorization code for tokens" to generate the access token.

3) Copy the generated access token and paste it here:

```
:=>ya29.AHES6ZRvf0GqfI4yv2LvZXGIF2eGyz34nymGpRk1L_4F0i9SFPsv1w
```

```
[1] "Your query matched 99 results that are stored to
↳dataframe ga.data"
>
```

The `ga.profiles` variable holds the following values:

```
> ga.profiles
      id                               name
1  725011      users.sch.gr/tsoukalos/
2  725056      www.lprotopapas.gr
3  2780821  gym-ag-anarg.att.sch.gr/library/
4  2814395      gym-ag-anarg.att.sch.gr/
5  5793223      store.kagi.com
6  5921572      widgetbook.blogspot.com/
7  21911813  tsoukalosphotography.blogspot.com
8  50079161      Truth Target
```

The returned values are all the supported Google Analytics profiles that I have in my Google Analytics account.

The important thing to remember here is that you can access your Google Analytics data natively from R. What you can do with the data is up to your imagination.

Using R for System Administration Purposes

This section describes how to extract useful information from a log file of an Apache Web server and analyze it using R. The name of the log file is `www6.ex000704.log`

and is hard-coded inside the shell script. You should change its name it to match yours.

A (small) shell script (called `www.sh`) is used to extract the preferred information from the Apache log file. Here's the script:

```
#!/bin/bash
echo "Time" "ServerBytes" "ClientBytes" "StatusCode"
grep -v '^#' www6.ex000704.log | awk '{print $2, $10, $11, $9}'
➔ | sed 's:// /g' |
awk '{print $1 ":" $2, $4, $5, "_"$6}'
```

The data is saved in a file called `www.data` using the following command:

```
$ ./www.sh > www.data
```

Here are the first ten lines of the `www.data` file so you can understand its format:

```
Time ServerBytes ClientBytes StatusCode
00:00 141 433 _304
00:00 142 437 _304
00:00 0 426 _200
00:00 142 435 _304
00:00 142 431 _304
00:00 114096 465 _200
00:00 141 436 _304
00:00 0 295 _200
00:00 141 434 _304
```

Note that the underscore in front of the status code was added by the `www.sh` script so that the `StatusCode` will not be considered a numeric value by R. The `read.table()` command is used to read the `www.data` file and import the data. Then the `summary()` command is used to get a general overview of the `WWWDATA` data set:

```
> WWWDATA <- read.table("./www.data", header=TRUE )
> summary(WWWDATA)
      Time      ServerBytes      ClientBytes      StatusCode
10:46 :    3145  Min.   :      0  Min.   :  0.0  _304   :709255
10:58 :    3081  1st Qu.:    140  1st Qu.: 401.0  _200   :435146
10:55 :    3066  Median :    142  Median : 435.0  _302   : 7371
10:37 :    3054  Mean   :   2460  Mean   : 438.1  _404   : 4641
10:32 :    2959  3rd Qu.:    407  3rd Qu.: 470.0  _500   : 3983
09:30 :    2814  Max.   :49083902  Max.   :2158.0  _206   : 2254
(Other):1144676                                (Other):  145
>
```

The following statistical definitions will help you better understand the output of the `summary()` command:

- **Min.:** the minimum value of the whole data set.
- **Median:** an element that divides the data set into two subsets (left and right subsets) with the same number of elements. If the data set has an odd number of elements, the Median is part

of the data set. On the other hand, if the data set has an even number of elements, the Median is the mean value of the two center elements of the data set.

- 1st Qu.: the 1st Quartile (q1) is a value that does not necessarily belong to the data set, with the property that, at most, 25% of the data set values are smaller than q1, and, at most, 75% of the data set values are bigger than q1. Or more simply, you can consider it as the Median value of the left-half subset of the sorted data set. If the number of elements of the data set is such that q1 does not belong to the data set, it is produced by interpolating the two values at the left (v) and the right (w) of its position to the sorted data set as: $q1 = 0.75 * v + 0.25 * w$.
- Mean: the mean value of the data set (the sum of all values divided by the number of the items in the data set).
- 3rd Qu.: the 3rd Quartile (q3) is a value not necessarily belonging to the data set, with the property that, at most, 75% of the data set values are smaller than q3, and, at

most, 25% of the data set values are larger than q3. Put simply, you can consider the 3rd Quartile as the Median of the right-half subset of the sorted data set. If the number of elements of the data set is such that q3 does not belong to the data set, it is produced by interpolation of the two values at the left (v) and the right (w) of its position to the sorted data set as: $q3 = 0.25 * v + 0.75 * w$.

- Max.: the maximum value found in the data set. Note that many practices exist for finding Quartiles. If you try another statistical package, you may get slightly different results.

The `summary()` command provides very useful information about the data set. Above, you can see that the busiest minute was 10:46 when 3145 requests were served. You also can see that there were 4641 “Not found” error messages (denoted by the 404 StatusCode number) out of a total of about 1.1 million page requests.

The `pairs()` command produces an impressive matrix of scatterplots—a scatterplot is a diagram that uses Cartesian coordinates to display values for two variables for a set of data. It

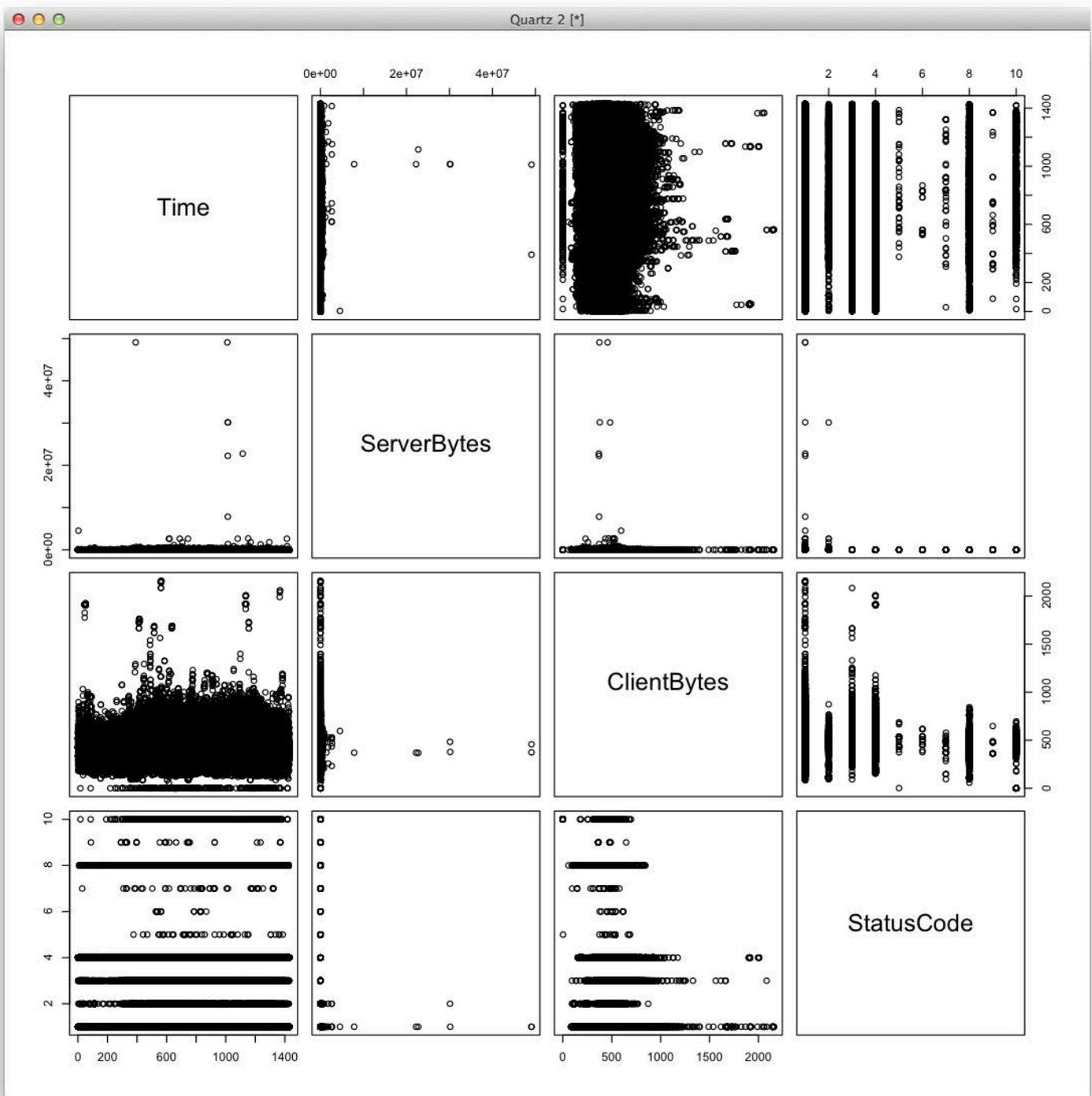


Figure 2. The pairs (WWWDATA) Command Output

helps you get a quick visual overview of your data:

```
> pairs(WWWDATA)
```

Figure 2 shows the output of the pairs() command, which is impressive! As WWWDATA is a large data set, I had to

R also can communicate natively with many database management systems.

wait a couple minutes for the `pairs(WWWDATA)` command to finish and produce its scatterplots.

Communicating with Databases

R also can communicate natively with many database management systems. For simplicity's sake, the database I use here is SQLite3; other popular supported options include MySQL, H2 and PostgreSQL.

SQLite is a public domain software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. Its main advantage is that it does not need a server process to run. Its main disadvantage is that, for the same reason, it cannot operate with multiple users.

Let's create an SQLite3 database (a single file) using R commands, and then import the WWWDATA data set inside an SQLite3 table. Commas are used to separate the different column values, so the WWW.sh file needs to to change a little.

R can communicate with an SQLite3

database in two ways:

1. Using the RSQLite CRAN package, which you can install using the `install.packages("RSQLite")` R command.
2. Using the sqldf CRAN package (sqldf makes use of RSQLite, so installing sqldf also installs RSQLite). You can install it using the `install.packages("sqldf")` R command.

Both packages need the DBI R package, which, as you easily can understand, will be installed automatically before installing either of them. This example uses the sqldf package. Loading the sqldf package with the `library()` command produces the following output:

```
> library(sqldf)
Loading required package: DBI
Loading required package: gsubfn
Loading required package: proto
Loading required namespace: tcltk
Loading required package: chron
Loading required package: RSQLite
Loading required package: RSQLite.extfuns
>
```

The slightly changed `www.sh` script, called `wwwcomma.sh`, is the following:

```
#!/bin/bash

echo "Time," "ServerBytes," "ClientBytes," "StatusCode"

grep -v '^#' www6.ex000704.log | awk '{print $2, $10, $11, $9}'
  ↪ | sed 's:/:/g' |

awk '{print $1 ":" $2",", $4",", $5",", $6}'
```

The data is saved in a file called `wwwcomma.data` using the following command:

```
$ ./wwwcomma.sh > wwwcomma.data
```

The R script (named `db.R`) that does the job is the following (the line numbers are added for clarity and need not be typed):

```
1. library(sqldf)
2. db <- dbConnect(SQLite(), dbname="WWW.sqlite")
3. wwwdatacomma <- read.csv("wwwcomma.data")
4. dbWriteTable(conn = db, name = "WWW", value = wwwdatacomma,
  ↪ RAW.NAMES=FALSE, APPEND=TRUE)
5. dbDisconnect(db)
```

Now, let's look at the R script line by line:

- 1: loads the required library.
- 2: creates a new database file

called `WWW.sqlite`.

- 3: after creating the database file, it reads the `wwwcommma.data` CSV file into R and saves it into the `wwwdatacomma` variable.
- 4: imports the data frames into the database in a table called `WWW`.
- 5: closes the db connection.

Additional handy commands include `dbListTables(db)`, which lists all the tables in a database using the `db` database connection; `dbListFields(db, "WWW")`, which lists all the fields of the `WWW` table using the `db` connection; and `dbReadTable(db, "WWW")`, which is like executing `Select * from WWW` using the `db` database connection. If your table is too populated, expect to see many lines of output.

You also can run SQL commands, such as the following, without opening a database connection by directly accessing the SQLite database file:

```
> q <- sqldf("SELECT count(*) FROM WWW", dbname = "WWW.sqlite")
Loading required package: tcltk

> q

  count(*)
1 1162795
```


SUBLIME TEXT: ONE EDITOR TO RULE THEM ALL?

**Get started editing
code like a pro,
with Sublime Text, a
programmer's editor.**

KEN KINDER

Sublime Text is a proprietary, cross-platform text editor designed for people who spend huge amounts of time shuffling code around. A programmer's editor, Sublime Text is a third option to the long-standing "Vi or Emacs" conundrum. Going beyond the basics of syntax highlighting and code folding, Sublime offers a litany of innovative and unique features. With version 3.0 just around the

corner, I'm taking you on a tour of Sublime's most compelling features and add-on packages.

At the time of this writing, Sublime Text version 2 is \$70 US, and the upgrade to version 3 (which is currently in beta) will be paid. Version 2 is downloadable as a trial, allowing you to get a feel for the editor for as long as you need before committing to buy. Because the application is available

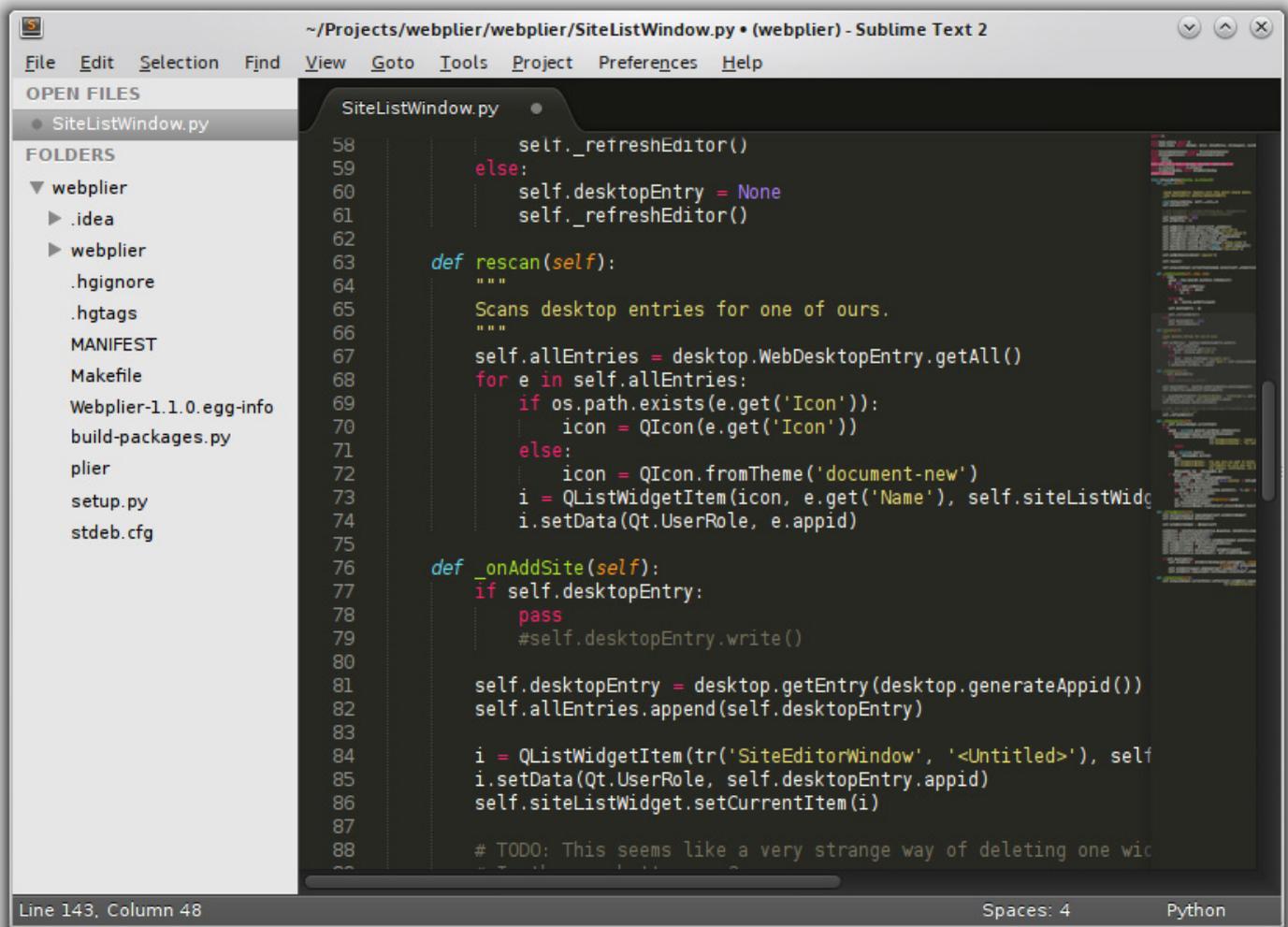


Figure 1. Sublime Text Editor Window

for Linux, Windows and Mac OS X, you do not need to buy a separate license for each platform. \$70 US may seem like a lot for a text editor, but if you spend hundreds of hours a month in front of your editor, it's a worthy investment.

Most of the content in this article should apply to either Sublime Text 2 or 3. Sublime Text 3 is not available for pre-purchase evaluation, so if you're new to Sublime Text, you'll be stuck with version 2 for now. You can download Sublime Text from <http://www.sublimetext.com>.

Getting Around in Sublime Text

Start Sublime Text, and the first thing you're greeted with is a charcoal editor window. A traditional project sidebar is on the left, and on the right, is what Sublime calls the Minimap. The Minimap is a zoomed-out view of the currently open file, which works a bit like a WYSIWYG scroll bar. Open some source code, and the Minimap provides a useful way of navigating large files visually.

If you have a directory holding a project to work on, choose File→Open Folder to select the project folder, then save the project by using Project→Save Project As. Consistent with the spirit of Sublime

Text, you can tweak the properties of the project simply by opening the `.sublime-project` file directly and editing its contents.

Open files in Sublime Text are shown in tabs reminiscent of Chrome. You can reorder and drag them between open Sublime Text windows, again like in Chrome or Firefox. This feature is particularly nice if you have multiple monitors, as it lets you quickly organize a vast workspace. If you want to focus (on writing a *Linux Journal* article, perhaps), use View→Enter Distraction Free Mode (Shift-F11) to view your file in full screen with all navigation widgets hidden.

Part of Sublime Text's appeal is speed, both in terms of application performance and UI design. A wide array of highly customizable keyboard shortcuts make using the mouse optional. My most frequently used hotkey is called Goto Anything and is available from the Goto→Goto Anything menu item (Ctrl-p). Provided you have the relevant language support installed (more on that later), Goto Anything lets you quickly access files, classes, functions and even regular old variables as you type. For example, I'll open up my project's `icongrabber.py` file by pressing

Ctrl-p, and as I type my desired filename, Sublime Text narrows down possible completions. When using Goto Anything, you can prefix your query with @ to find a symbol, # to search within a file or : to jump to a line number. Unfortunately, Sublime Text does not search symbols in unopened files.

A close second in useful keyboard shortcuts is the Command Palette.

Similar to the Escape/Command prompt in Emacs, Sublime's Command Palette lets you quickly execute commands internal to Sublime Text or provided by an add-on package you've installed. For example, to toggle word wrap, use Tools→Command Palette (Shift-Ctrl-p) and type "wrap". Sublime is smart enough to suggest "Toggle Word Wrap" as a completion.

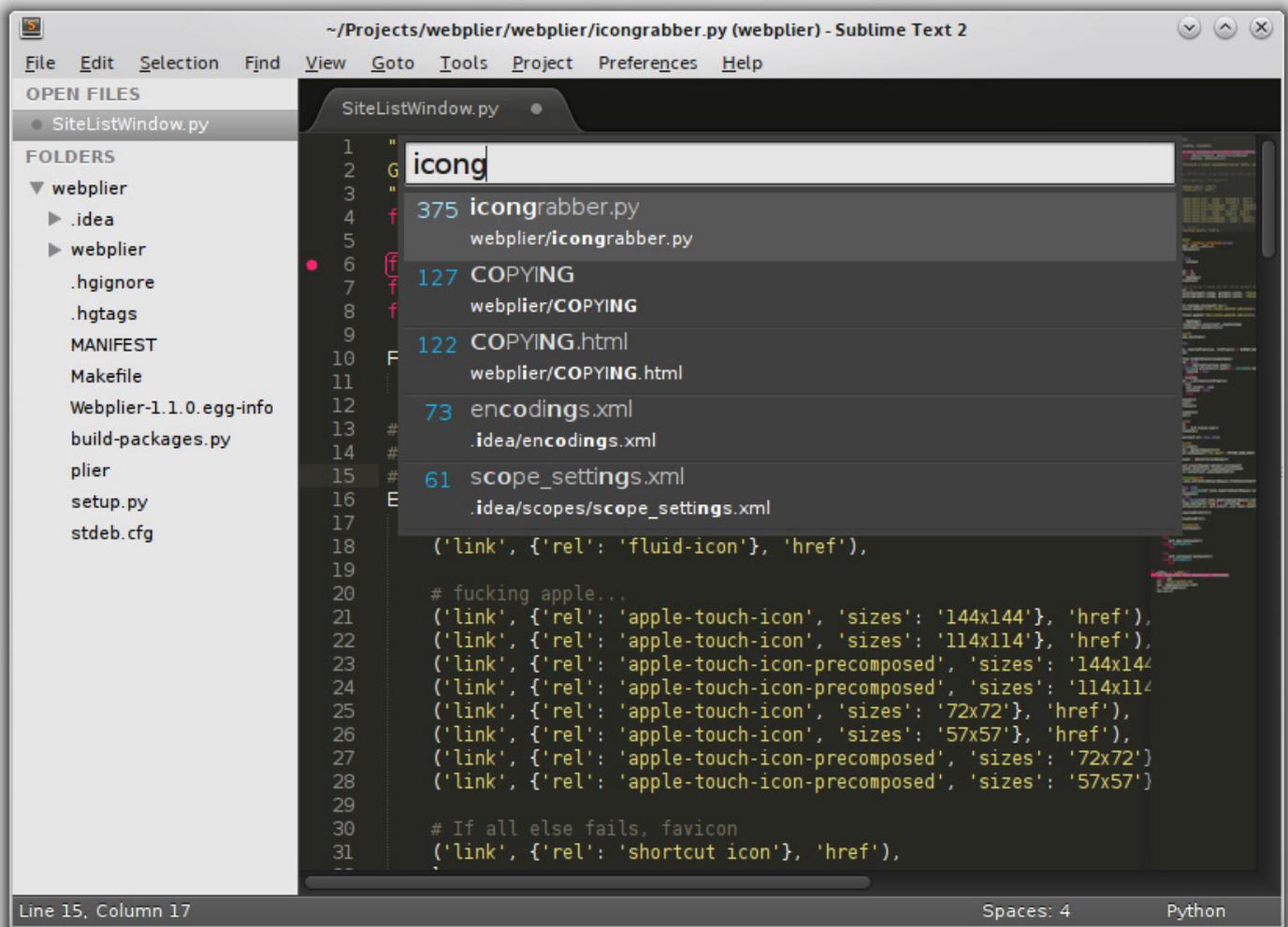


Figure 2. As you type the name of your file, Sublime Text narrows down possible completions.

SUBLIME TEXT LETS YOU SELECT MULTIPLE NONCONTIGUOUS SPANS OF TEXT AND ACT ON THEM COLLECTIVELY.

Notice that Sublime Text also shows keyboard shortcuts for commands that have them.

To view a full list of default key bindings, click on the Preferences Menu and choose “Key Bindings - Default”. This will open up the system-wide key-binding file. To create your own key-binding preferences, choose “Key Binding - User”, and use the same syntax as the default file.

Editing Kung Fu

Now to the heart of what makes Sublime Text such a powerful editor: its unique alchemy of text editing features. Sublime Text’s most praised editing feature is multi-selection, which is a little tricky to wrap your head around at first. Most editors let you select only one contiguous span of text; some let you select text as a block. Sublime Text lets you select multiple noncontiguous spans of text and act on them collectively. After you’ve begun using this feature, its power will become apparent to you, especially in editing code or any file

with a formal syntax.

Say, for example, that I’m converting the following source code from Python 2 to 3. The first thing I want to do is rename “raw_input” to just “input”:

```
your_name = raw_input('Enter your name: ')
print 'Hello,', your_name

printer_model = raw_input("What kind of printer do you have?: ")
print your_name, 'has a', printer_model
```

Using Sublime Text, such a task is easy. I’ll select the first occurrence of “raw_input” and press Ctrl-d. Pay close attention, and you’ll notice that both occurrences of raw_input are now selected, each with its own blinking cursor. As I begin to type the word “input”, both occurrences are replaced. It is true that such a change could have been accomplished easily with search and replace, but I’ve only scratched the surface with multiple selection.

Next, I’ll want to replace the two “print” statements with Python 3’s “print” function, which means making the commands look like `print(...)`.

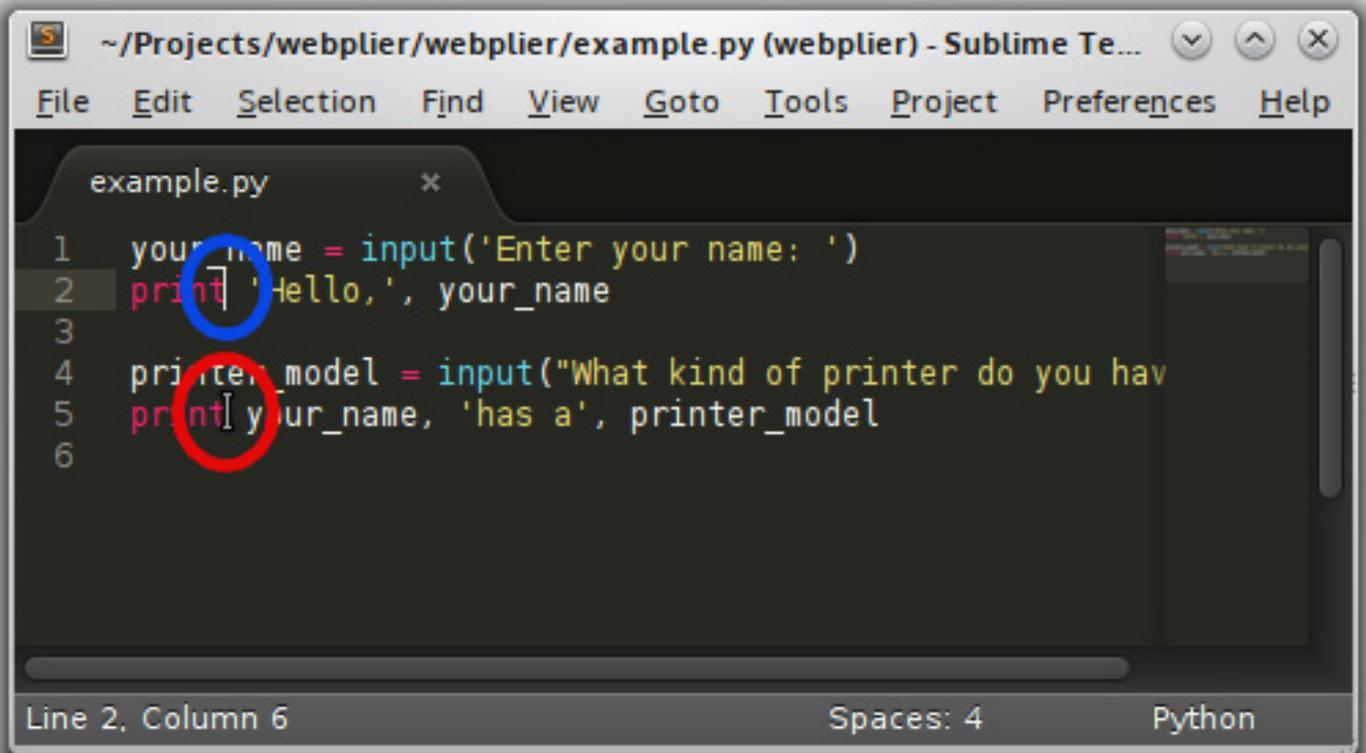


Figure 3. The editor cursor is blue, and the mouse pointer is red. By holding down Ctrl and clicking, you can create multiple editor cursors.

Because the text “print” occurs four times in this document, the last technique won’t work, so I’ll show you another way to make multiple selections. I’ll begin by positioning my cursor on the first print statement. Then, I’ll hold Ctrl while clicking on the other print statement.

Although I haven’t selected any text, I have two blinking cursors. Whatever I type will effect both lines. I’ll type (, press end, and type). Both lines received those keys, and now my file is Python 3-compliant.

There are several other ways of selecting multiple spans of text in

Sublime Text, and as you experiment with them, you’ll get a feel for how to use them. Ultimately, when used productively, Sublime’s multiple selection feature replaces most editor macro, find/replace operations and refactoring operations all at once. Imagine, for example, how you could transform a plain-text list into an HTML list using multiple selection. If you can picture how this is done, you’re starting to grok multiple selection. If not, don’t fret: this is a new editor feature. A few trips to YouTube and some video demos will help you get the idea.

Search and Replace

Forget grepping through your codebase when the time comes for aggressive refactoring. Sublime Text offers a powerful recursive search and replace feature. Recursive search and replace eliminates the need for the GNU `grep` and `find` commands for many users. Many

editors provide recursive search and replace, although I find Sublime Text really gets it right in a way that few other projects do.

Click on `Find`→`Find in Files`, and a large search bar will appear at the bottom of your editor. Using the toggle buttons on the left, you can toggle regular expression matching,

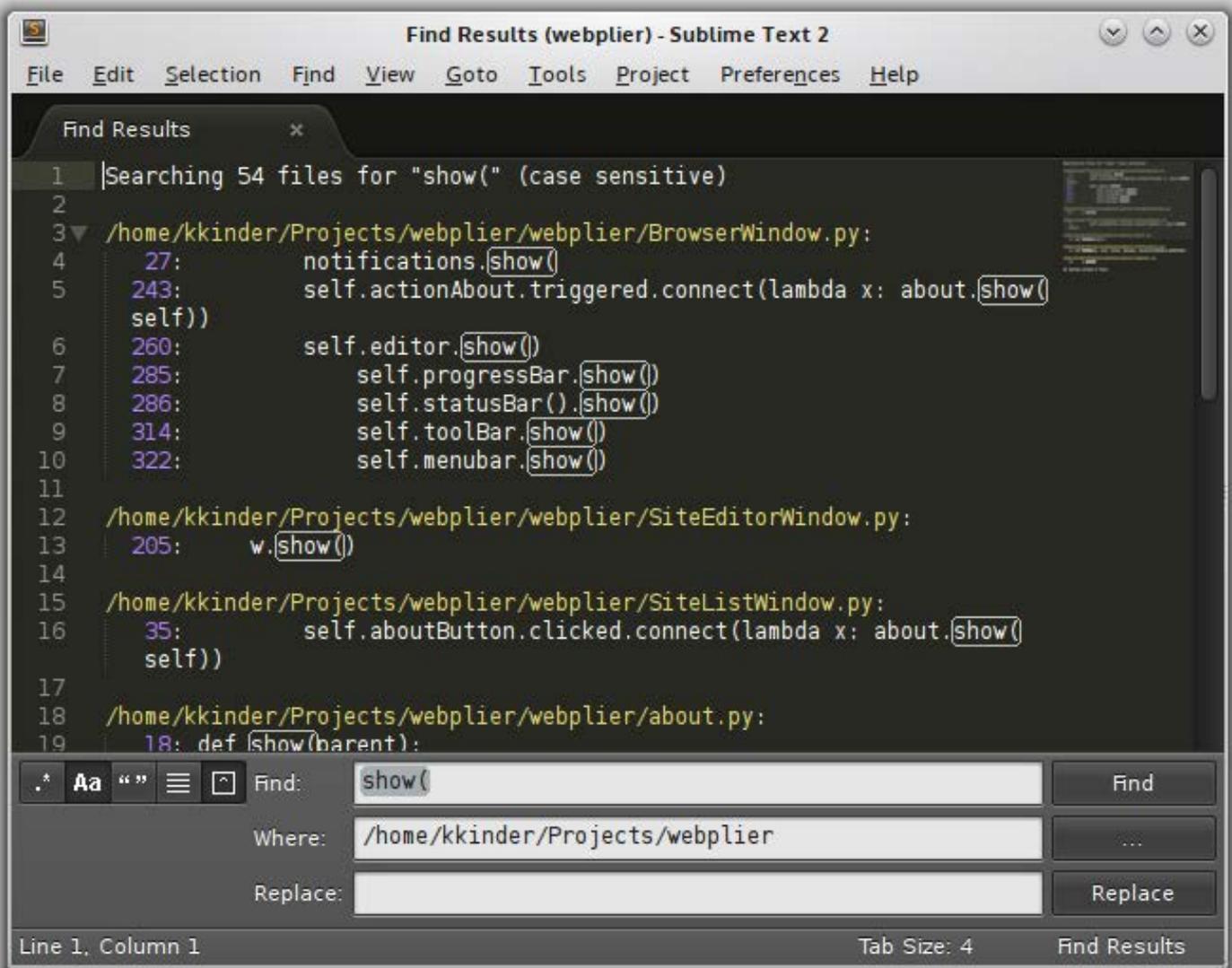


Figure 4. Sublime Text can open up your search results in its own buffer. Click on any result to jump to its source.

case sensitivity and whole words. Hover over individual icons to see what they do. You also can select a directory to search and optionally specify replacement text.

If you expect many results or plan to refer to your search results over time, toggle the “Use Buffer” icon in the search area. When enabled, Sublime Text will open a summary of search results in its own editor buffer. When using a multihead workstation, I find it useful to put search results in one monitor and code in another. Toggling “Show Context” will include a few lines before and after each hit in the results.

Sublime Text uses Perl-style regular expressions implemented using the Boost C++ library. Sublime Text also supports regular expression replacements.

Snippets

Despite their best efforts to not repeat themselves, programmers often find themselves typing common blocks of text throughout their projects. Examples include standard class file layouts, unit tests and license warnings programmers put at the top of each file.

To support this work flow, Sublime Text features “snippets”. Suppose I have a standard unit test layout for

my Python projects:

```
"""
Unit tests for <MODULE> in <PROJECT>.
"""

import unittest

class UnitTest(unittest.TestCase):

    def setUp(self):
        """
        Called before each test to set up the environment.
        """
        pass

    def tearDown(self):
        """
        Called after each test to clean up.
        """
        pass

    def test<METHOD>(self):
        pass

if __name__ == '__main__':
    unittest.main()
```

I can use Tools→New Snippet. Sublime Text will give me an example snippet file. I’ll modify it to read like this:

```
<snippet>
    <content><![CDATA[
"""
Unit tests for ${1:module} in ${1:project}.

```

I LIKE TO THINK THAT SUBLIME WALKS THE FINE LINE BETWEEN AN IDE AND A TEXT EDITOR.

```
"""
import unittest

class UnitTest(unittest.TestCase):

    def setUp(self):
        """
        Called before each test to set up the environment.
        """
        pass

    def tearDown(self):
        """
        Called after each test to clean up.
        """
        pass

    def test${1:method}(self):
        pass

if __name__ == '__main__':
    unittest.main()

]]></content>
<tabTrigger>unittest</tabTrigger>
<scope>source.python</scope>
</snippet>
```

Notice that I've made a "tabTrigger" tag and a "scope"

tag in my snippet file. Using the settings I've given, any time I'm in a Python file and type the word "unittest", I can press Tab, and the snippet will be inserted where the cursor is. To try it out, I'll save the snippet as "unittest.sublime-snippet" in the default directory. Now I can use the snippet to create unit tests quickly.

Packages Galore

Sublime Text can be scripted using plugins written in Python. These plugins are stored in packages that can be installed locally using a file manager or your favorite shell. Should you feel the urge to scratch an itch no one else has found, you can write packages in Python (more on that later).

I like to think that Sublime walks the fine line between an IDE and a text editor. Its speed and initial simplicity make it suitable for editing /etc files as easily as source code. Sublime's true power, however, is found in add-on packages users can write and install to do everything from synchronizing

files over SSH to refactoring code. Basic syntax highlighting is included for major languages, but to make use of Python as a programming tool, I find it's best to install some handy packages.

Before anything else, you'll most likely want to install something called Package Control, which is a little bit like apt-get for Sublime Text. Package Control is itself a package that manages downloading, installing, updating and removing other packages. Download Package Control from http://wbond.net/sublime_packages/package_control. To install a package, just use the Tools→Command Palette (Shift-Ctrl-P), and type "Package Control". Along with other actions, "Install Package" will be available as a completion.

Finding Coding Errors

One of the most compelling features an IDE offers over a text editor is real-time error detection. For example, if you type a Java syntax error into Eclipse, the editor realizes your mistake and warns you about it in real time.

SublimeLinter provides similar functionality for a variety of languages inside Sublime Text. Install

SublimeLinter using the Package Control "Install Package" command described above or by downloading it from <https://github.com/SublimeLinter/SublimeLinter>.

SublimeLinter wraps native language tools, such as cppcheck for C and xmllint for XML, so you'll need the relevant tool installed for your language. Let's try an XML error.

Can you spot the error in Figure 5? Notice in the gutter of the editor, there's a warning icon. You'll see in the status bar, SublimeLint is explaining the problem: I forgot to close the <head> tag. After fixing the problem, pressing Ctrl-Shift-L will force SublimeLint immediately to rescan the file, and the error will go away.

For Python Programmers

If you're a Python programmer, your first download for Python Develop undoubtedly will be SublimeRope. SublimeRope combines Python's Rope source code analysis and refactoring library with Sublime Text, offering context-specific completion, refactoring, jumping to symbols using Sublime Text's "Goto Anything" feature and more.

Install SublimeRope by using the Package Control "Install Package" command. To test out just one of

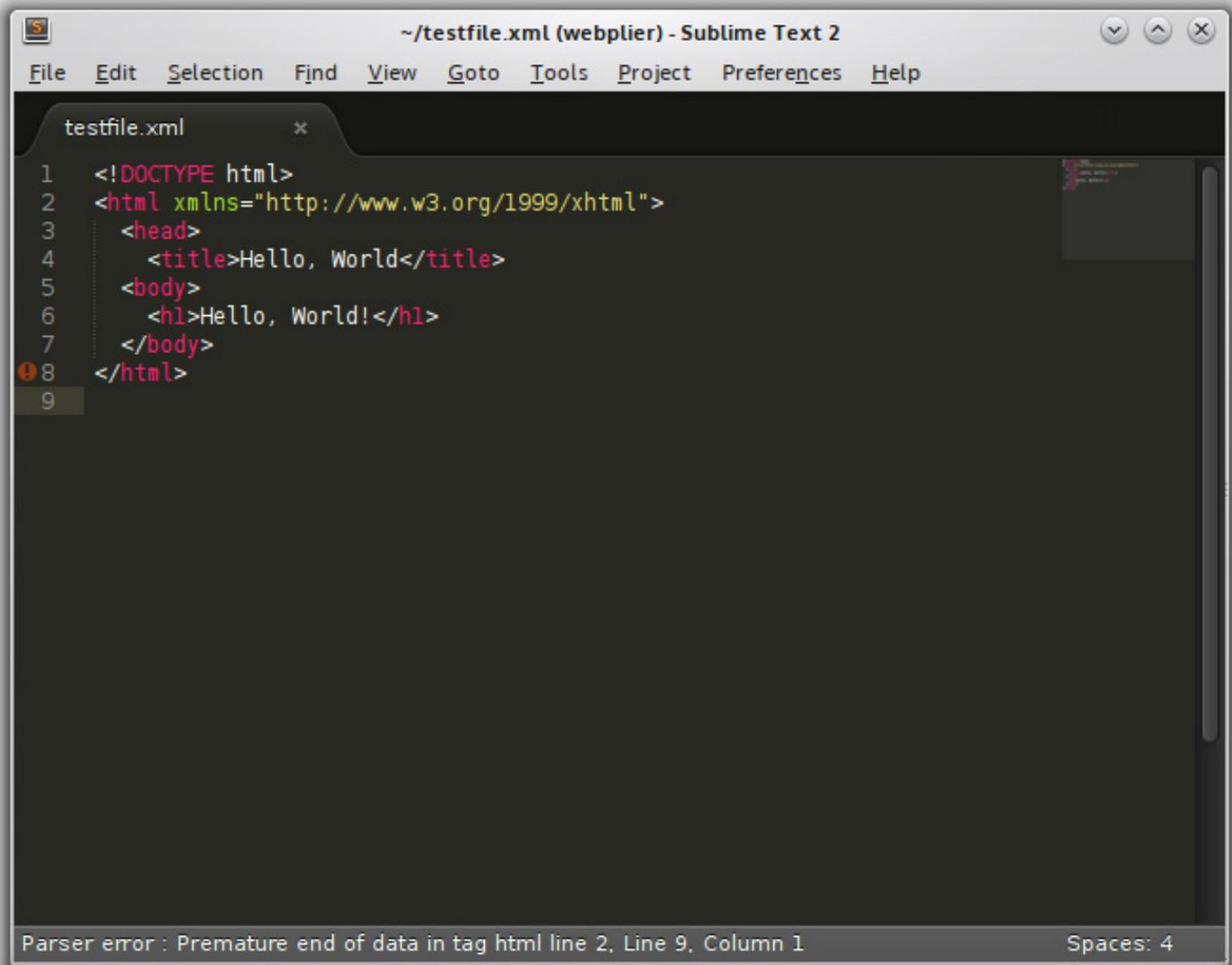


Figure 5. Notice that Sublime Text highlights lines with errors, and in the status bar, it describes the error itself.

Sublime Rope's features, try this code:

```
#!/usr/bin/env python2.7

def hello(name):
    print 'Hello, %s' % name

hello(raw_input('Enter your name: '))
```

Move the cursor over the definition of the hello function. To use a SublimeRope command to rename "hello" to "greet", use the Command Palette (Shift-Ctrl-P) and type "rename". You should notice a "Rope Refactoring: Rename" command. After choosing the Rename command, enter "greeting" as the new name

of your function, and notice that the name has been replaced in both places.

To explore other features of SublimeRope, including organizing imports and showing documentation of Python methods, just use the Command Palette and type “rope” to see the handful of commands SublimeRope provides. In general, this is a quick way to explore commands provided by packages you add to Sublime Text.

Synchronizing Code to a Server

As a Web developer, I find myself testing and developing code on servers. Although one way to do this would be to make changes locally and rsync them up to the server after each edit, with a large codebase, this is a painfully slow solution. Another option is to use sshfs to mount a remote filesystem locally, but this too has its problems, especially in terms of latency over a typical broadband connection.

Enter Sublime SFTP. Although Sublime SFTP is a \$16 “shareware” package, like Sublime Text itself, SFTP support is a justified expense for anyone who uses Sublime Text for a living. At the time of this writing, Sublime SFTP is available only for Sublime Text 2. Install Sublime FTP using the same method as other packages. Use Package

Control’s “Install Package” command and find the package called just “SFTP” in the list.

To get started, choose File→SFTP/FTP→Setup Server. Sublime Text will open up a file letting you specify a hostname, user name and so on. Sublime SFTP will use your default SSH keys, so if you’ve already configured logging in to your remote host, this will be easy. Settings for remote servers are stored as files in ~/.config/sublime-text-2/Packages/User/sftp_servers. Each file in this directory represents a remote server, and files can be directly manipulated to update settings.

After configuring a server, you can open files remotely by going to File→SFTP/FTP→Browse Server or mapping a local directory to be synchronized remotely by right-clicking on a directory in your project and choosing SFTP/FTP→Map to Remote.

In my experience as a developer, Sublime SFTP is a surprisingly reliable and well-made package, well worth its unusually high price.

Where to Go from Here

Of course, I’ve just scratched the surface of Sublime Text’s capabilities in this article. I’ve glossed over or left out many great features, and as you use Sublime Text, you’ll find a great deal more. If I’ve whet your appetite

Why Join USENIX?

We support members' professional and technical development through many ongoing activities, including:

- » Open access to research presented at our events
- » Workshops on hot topics
- » Conferences presenting the latest in research and practice
- » LISA: The USENIX Special Interest Group for Sysadmins
- » *;login.*, the magazine of USENIX
- » Student outreach

Your membership dollars go towards programs including:

- » Open access policy: All conference papers and videos are immediately free to everyone upon publication
- » Student program, including grants for conference attendance
- » Good Works program

Helping our many communities share, develop, and adopt ground-breaking ideas in advanced technology

Join us at www.usenix.org



usenix

THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

GNU Awk 4.1

Teaching an Old Bird Some New Tricks, Part II

gawk 4.1 lets you use really big numbers, and finally talk to your OS.

ARNOLD ROBBINS

In an earlier article (“GNU Awk 4.0: Teaching an Old Bird Some New Tricks”, published in the September 2011 issue of *Linux Journal*; see Resources), I gave a brief history of awk and gawk and provided a high-level overview of the many new features in gawk 4.0. I recommend reading that article first, although you can read this one without doing so, if you wish.

gawk 4.0 itself was released in June 2011. Since then, the gawk development team has not been resting on its laurels! gawk 4.1, released in May 2013, contains a number of new features, and that’s what I cover here.

Unlike gawk 4.0, there are considerably fewer changes at the language level (although there are some). The changes this time around are more concerned with internals, and with the ability to interface to the outside world. So let’s get started.

Reduced Footprint

For many years, when you built gawk, you got two executables: the regular interpreter, gawk, and pgawk, its profiling twin brother, which ran awk programs (more slowly) and produced a statement count execution profile showing how many times each line of

code was executed.

With gawk 4.0, you got an additional executable, dgawk, the gawk debugger. Although the three versions shared most of the same code, the core parts that actually executed your awk program were compiled differently in each one.

For gawk 4.1, all three executables have been merged into a single program, named just gawk. Although the combined executable is larger, it is still smaller than having three separate executables, and in addition, the documentation is simpler and easier to understand (and maintain!).

To accommodate this change, the options had to change slightly. You now use `-D` to run the debugger, `-p` to do profiling and `-o` for pretty-printing without profiling.

Arbitrary Precision Arithmetic with MPFR and GMP

An important new feature that is visible for the awk programmer is arbitrary precision floating-point arithmetic with the GNU MPFR and GMP libraries.

This is an optional feature: if you have the MPFR and GMP libraries installed when you configure and build gawk, gawk automatically will be able to use them.

Note that I said “be able to use them”. You still have to choose to do so either by using the `-M` option (or `--bignum`, if you prefer long options), or by setting the special variable `PREC` to the desired floating-point precision.

The precision is the number of bits kept in the floating-point mantissa. The default is 53, which is the same as that used by hardware double-precision floating point. From the `gawk` manual:

```
$ gawk -M -v PREC=100 'BEGIN { x = 1.0e-400; print x + 0 }
> PREC = "double"; print x + 0 }'
1e-400
0
```

You see that regular hardware can't handle an exponent of `-400`, whereas MPFR can.

An additional new variable, `ROUNDMODE`, sets the rounding mode for calculations and printing arbitrary precision values.

In the past several years, for reasons I don't quite understand, I've gotten bug reports from people who expect `gawk`'s arithmetic to work exactly like “real” arithmetic done with pencil and paper. In other words, they want what is known in Computer Science as decimal arithmetic. I'm not sure why they expect this, but as we all should

know, computers don't quite work that way.

MPFR does not give you decimal arithmetic. However, if you understand what you're doing and how to use it, you can get results that are likely to be good enough for your purposes.

The manual has a full chapter that describes the issues involved with floating-point arithmetic, what it means when you increase the precision, and how to use the various rounding modes supported by MPFR.

New Arrays Provide Indirect Variable Access

There are three new arrays:

- `SYMTAB`: provides access to awk-level variables.
- `FUNCTAB`: lists the names of all user-defined and extension functions.
- `PROCINFO["identifiers"]`: lists all known identifiers and what `gawk` knows about their types after it has parsed the program.

Of these, `SYMTAB` is the most interesting, since it provides indirect

By contrast, modern scripting languages are all open and extensible; Perl, Tcl, Python and Ruby all have thousands of available modules that can be loaded at runtime. It's past time that gawk could do that too.

access to any variable. For example:

```
$ gawk 'BEGIN { a = 5 ; print "a =", a
> SYMTAB["a"] += 37
> print "a is now", a }'
a = 5
a is now 42
```

With the `isarray()` built-in function, you can “walk” the entire symbol table and print out all variable and array values, if you choose to do so.

Dynamic Extensions

The most exciting change in gawk 4.1 is its ability to interface to the outside world. For many years, gawk had an “extension” or “plug in” mechanism that let a programmer write a new “built-in” function in C, and load it into the running gawk interpreter at runtime.

This mechanism required understanding something of the gawk internals and making use of gawk's

internal data structures and functions. Although it was documented minimally, and it worked, it had several drawbacks. The most notable one was that there was no backward compatibility across releases.

Nonetheless, a group of developers forked gawk to create xgawk (XML gawk) and developed a number of dynamic extensions and new facilities for the core executable.

For many years, I had been wanting to provide a defined C API for writing extensions that would not be dependent upon the gawk internals and that possibly could provide binary compatibility across releases.

For gawk 4.1, together with the xgawk developers, we finally made this happen.

Why Do You Need Extensions?

Consider this: an awk program cannot even change its working directory with the `chdir` system call! awk is thus a closed language—one that provides you with only

the facilities that the implementors chose to provide and no more. That's not much fun. (Well, awk is fun, but it's still limited.)

By contrast, modern scripting languages are all open and extensible; Perl, Tcl, Python and Ruby all have thousands of available modules that can be loaded at runtime. It's past time that gawk could do that too.

What You Can Do from an Extension It is best to think of extension functions as user-defined functions written in another language. They cannot do everything a user-defined function can (such as call an awk function, manipulate the fields, read records with `getline` and so on), but what they can do is enough to make gawk more open, and let it interface with the underlying operating system and with other C (or C++) libraries. In particular, you can:

- Pass scalars by value and arrays by reference.
- Create and modify new global variables and arrays.
- Access the built-in variables (read-only, although you can update `PROCINFO`).

- Register a function to be called when gawk exits.
- Print warning and/or fatal error messages.
- Update the built-in variable `ERRNO` for when something goes wrong.
- Hook into the I/O redirection mechanisms, providing your own "special" filenames and/or two-way communicators.
- And of course, register new functions that can be called from gawk.

The API provides a number of data types to make it easier to communicate with gawk. For example, gawk strings can contain embedded NUL characters (all bits zero), so strings have a pointer and a length. gawk maintains reference-counted strings internally, so there are ways to tell gawk to reuse a value it already knows about.

In addition, the API lets you "flatten" awk's associative arrays into an array of structs for easy iteration in C code, without having to call into gawk each time you want to move to the next element in an array.

A full description of the API is beyond the scope of this article; however, the manual includes a full chapter, with examples, describing the API and showing how to use it.

OS Independence The extension mechanism has been designed to work on multiple operating systems. At the time of this writing, it works on any *nix system that supports the POSIX `dlopen()` API. This includes Mac OS X. The basic mechanism also works on Microsoft Windows using MinGW. However, support to build the sample extensions was not included in the 4.1 release since it was not ready. This support will be included in the first patch release, whenever that will be, although not all of the sample extensions can work on Windows.

Sample Extensions The `gawk` distribution provides a number of small, sample extensions. Their main purpose is to serve as examples of how to use the API, but nonetheless they should be usable for real work also. The full list is documented in the manual. Some of the more interesting ones are:

- The “filefuncs” extension, which provides `chdir()` and `stat()` functions, and also an interface to the `fts(3)` suite of routines for walking a file hierarchy.

- The “fnmatch” extension, which provides an `awk` version of the `fnmatch(3)` suite.
- The “readdir” extension, which returns records for the contents of directories named on the `gawk` command line or read with `getline`. (Normally, it’s a nonfatal error to try to read a directory. With other `awks`, it’s fatal.)
- The “inplace” extension, which simulates the GNU `sed -i` feature for in-place editing of command-line data files.

Additional, more specialized extensions illustrate the use of parts of the API not covered by the extensions just listed.

The `gawkextlib` Project Now that `gawk` supports the major `xgawk` features, the `xgawk` developers have reoriented their project around their specific extensions. It no longer includes the forked `gawk` code base. To emphasize this change in orientation, they renamed their project “`gawkextlib`”.

It is their (and my) hope that this project can serve as a central clearinghouse for new `gawk` extensions that may be written by the

awk community over time.

The gawkextlib project currently has four extensions:

- The XML extension, which adds several new variables and an input parser, letting gawk parse XML files in a natural fashion. This extension is built on top of the Expat XML parser. This is a powerful extension; instead of having to try to parse XML files with regular expressions manually, the Expat parser does it for you, including all the icky validation stuff that would be really hard to do in straight awk code.
- The PostgreSQL extension, which provides functions for talking to PostgreSQL databases.
- The GD graphics library extension, for use with the GD graphics library (see Resources).
- The MPFR library extension. This extension gives you access to a number of MPFR functions that are not accessible from gawk's built-in MPFR support.

The Future

I feel that gawk as a language has largely reached maturity, and do not

wish to add too many more features.

That said, there are a few items still open for exploration:

- Additional numeric facilities, such as possible integration with a decimal arithmetic library.
- A way to map gawk arrays onto external storage, such as DBM arrays or SQL databases.
- A “namespace” facility for extension functions and variables, and possibly regular gawk-level variables and functions as well. This would be a major design activity.

Of course, describing the above items does not constitute a commitment to do any of them.

Conclusion

The new API and extension facility opens new horizons for gawk and for awk programmers. I am very excited about it, and I hope to see gawk used for many new things where it simply was not applicable before.

Acknowledgements

Thanks to Scott Deifik, Dr Brian W. Kernighan, Dr Nelson Beebe and Eli Zaretskii for comments on the initial

draft of this article.

The entire gawk development team deserves kudos for their work on this release. It was very much a team effort. ■

Arnold Robbins is a programmer, technical author, husband and father. A native of Atlanta, Georgia ("American by birth, Southern by the grace of G-d"), he and his family have been living in Israel since 1997, where he now works writing software for a very large semiconductor manufacturing company. He has been involved with GNU Awk since 1987(!). In his non-copious spare time, he maintains gawk and its documentation, among

other activities. As a result of his involvement with gawk, he has had the privilege of meeting Brian Kernighan in person, who was kind enough to autograph Arnold's copies of all his books, including the *The Awk Programming Language*. Arnold is also the author or co-author of a number of UNIX- and Linux- related books from O'Reilly and Prentice Hall, which he hopes that all readers of this article will now run out and buy. For more information, see his home page at <http://www.skeeve.com>.

|||||
Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

Resources

"GNU Awk 4.0: Teaching an Old Bird Some New Tricks", *LJ*, September 2011:
<http://www.linuxjournaldigital.com/linuxjournal/201109#pg94>

The gawk distribution: <ftp://ftp.gnu.org/gnu/gawk/gawk-4.1.0.tar.gz>

Documentation On-line: <http://www.gnu.org/software/gawk/manual>

Arbitrary Precision Arithmetic with gawk: http://www.gnu.org/software/gawk/manual/html_node/Arbitrary-Precision-Arithmetic.html#Arbitrary-Precision-Arithmetic

Dynamic Extensions: http://www.gnu.org/software/gawk/manual/html_node/Dynamic-Extensions.html#Dynamic-Extensions

gawkextlib Home Page: <http://gawkextlib.sourceforge.net>

gawkextlib Download: <http://sourceforge.net/projects/gawkextlib>

The GD Graphics Library: <http://www.boutell.com/gd/manual2.0.33.html>

The Expat XML Parser: <http://expat.sourceforge.net>

Get More Juice out of Your Enterprise Code Base with Code Search

Extract the wealth of knowledge trapped inside your code base.

SUSHIL KRISHNA BAJRACHARYA

When most people think about a company's reusable assets, source code doesn't usually show up on the list, even though millions of dollars are spent every year on creating and maintaining code. Most large companies are managing hundreds of millions of lines of code—the majority of which was purpose-built to solve a specific application problem. Most of that code is locked up in source control management systems (SCMs) specific to an application or a siloed organization.

Add to this the world of

open-source software development where similarly billions of lines of code exist, but where source code is shared publicly and regularly reused—both wholesale and through forking. Here too, plenty of effort and resources are spent in writing and maintaining source code. Source code is maintained, extended and reused by a large number of developers. And, like enterprise code, open-source code also is stored in various source code repositories.

Collectively, the code that lives in internal SCMs across large

organizations, together with the billions of lines of code that exist in the open-source world, reflect the implementation artifacts of literally millions of developers. These artifacts can be used as powerful resources to assist with the design, development, analysis and problem-solving of future applications.

But, how can we leverage this massive resource?

Code Search Engine

A code search engine is a tool that can help developers unlock the wealth of diverse implementation knowledge buried inside large repositories. A code search engine facilitates search operations that are specific to source code and applies analysis and heuristics specific to source code while processing, indexing and retrieving source code. A source code engine, unlike general text-based search engines, is designed and implemented especially to cater to developers' information needs related to source code. With these features, a code search engine facilitates source code search.

Code Search

Source code search (or simply, code search) is a technique to find relevant source code in multiple source code

repositories. Code search can help fulfill commonly occurring search needs during development tasks, such as finding the usage of APIs across different projects, finding how a known information structure is implemented in code (such as base 64 encoding) and so on. What a developer finds useful in code search results depends on the search need at hand. An effective code search engine facilitates fulfilling such search needs by delivering relevant results and providing the means to explore and narrow down search results in cases where the need is vague and unclear. Given that alternative choices are available in the results, a code search engine can act as a choice engine by allowing code-specific faceting and filtering mechanisms.

Enterprise Code Search

Enterprise code search is code search as applied inside a company's firewall, searching corporate source code repositories. Enterprise code search must adhere to additional enterprise requirements, such as authorization and access policies on source code visibility. This poses additional requirements and challenges when considering a code search engine for an enterprise, since the search tool has to meet the company's standards

Developers are not always looking for exact lines of code to copy and reuse. More often they seek useful patterns they can add to their repertoire of knowledge to solve recurring tasks.

and needs to fit with existing IT, enterprise tools and deployment procedures in place.

Use Cases for Code Search

Developers frequently use code search tools for copy-paste programming. Best practices developers frequently seek to reuse existing solutions, and once implemented, a common solution to a problem (such as a well-known algorithm) can be used again and again. Copying and pasting code from an existing solution, when legally permissible, often can be the most efficient approach, saving developers time and resources to focus on more challenging tasks. A code search engine can be an ideal tool to find such solutions. Although there certainly are reservations against practices like copy-paste programming, some of which are reasonable (for example, one might not be able to trust someone else's code blindly), code search engines deployed inside enterprises can winnow down results to internal projects that reveal code written by experts, helping

to alleviate such concerns while still permitting the much-practiced copy-paste programming.

Developers are not always looking for exact lines of code to copy and reuse. More often they seek useful patterns they can add to their repertoire of knowledge to solve recurring tasks. For example, while using APIs, developers need to learn the patterns of API usage. Today's applications frequently leverage API calls to other internal or external components. The typical API has little documentation and few good examples, so it can be frustrating and time-consuming for developers to figure out how to use them successfully. Two easy answers to this problem would be either to enable developers to see examples of how other developers have used an API or to provide visibility into the code behind the API. To accomplish this, developers need an easy way to search and view an API call or other code that calls the API. A code search engine allows developers to

accomplish this task easily. Code samples and examples are vital learning tools for developers who often will copy and modify existing examples to fit their purposes. A code search engine lets developers use existing code repositories as sources of examples—in the above case, sources of API usage examples.

A code search engine can be helpful in various other scenarios. When starting a new project with new languages and frameworks, developers would benefit by researching and studying the code bases of mature projects using the same languages and frameworks. Open-source implementations can be a great way for developers to learn solutions to complex computing problems, such as implementing distributed systems, search engines, network servers and so on. Code search engines in the enterprise also can be extremely helpful during normal development activities, such as maintenance, porting and working with legacy code. Code search engines can be used to index and cross-link files spanning multiple types and languages, thus supporting traceability in the search results. Developers can use code search during maintenance to find source files, unit tests and configuration files related to a particular feature.

Challenges in Code Search

The use cases presented above demonstrate the potential benefits of a code search engine, but these benefits cannot be realized unless the code search engine is effective and efficient. Code search results must be relevant, comprehensive and meet the users' information need for the tool to be effective. It must be designed with the features and capabilities needed by a wide range of developers who are under constant pressure to work more efficiently and cost effectively. To be efficient, the code search solution must be capable of delivering effective results within acceptable response times by having the capacity to scale to very large repositories.

Source code, unlike plain or natural language text, tends to be very sparse. This poses a serious challenge in building effective code search engines if one resorts only to techniques that work for natural text. The lack of rich vocabulary in code has to be compensated with additional attributes that can be leveraged and would exist only in source code. One such attribute is the rich structural information that exists in source code. Unlike natural text, source code is highly structured with definitions of various nested elements and relations between these elements. For example, in a typical

object-oriented program, one would find classes and methods, where classes extend to other classes, and method calls to other methods. A code search engine needs to parse source code to extract such elements to provide search operators that specifically allow the retrieval of these elements. For example, when a developer needs to find a certain method name, an operator (such as `mdef` in `ohloh.code`) easily can deliver effective search results on such a query.

This rich interlinked structure relates several elements with one another and can be the basis of accumulating similar terms when vocabulary is sparse. Similar to the Web, the link structure in code itself can be used to build new metrics of popularity and ranking, if used properly. There are several conventions (such as naming conventions) found in source code writing that are uncommon in natural text that make special tokenization and processing suitable for source code. (To learn more about these topics, refer to the author's doctoral dissertation: *Facilitating Internet-Scale Code Retrieval* at <http://dl.acm.org/citation.cfm?id=2019966>.)

For proper extraction of elements in source code and relations among such elements, a code search engine first needs to be able to detect

the implementation language and perform detailed parsing of the code, which can be nontrivial for complex languages and for repositories where erroneous or incomplete code exists.

Beyond lexical and structural properties, source code has executable properties making it an executable artifact with runtime behaviors that change as the code evolves. Understanding such behavior is vital to activities like fixing bugs or improving performance. A code search engine can leverage the stored representations of runtime behavior as captured in test coverage reports, call traces, profiling outputs and logs, and relate them with appropriate elements defined in source code to provide answers related to unexpected behavior in code.

Finally, being produced and maintained by developers who work collaboratively, source code even has human-centric attributes. Since most of the activities on source code are logged in source repositories, a source code engine can tap into information connected to such activities to provide answers related to developers and their activities when needed. For example, it can help find an expert on a certain feature, or a developer tasked with managing a specific project can be notified when a certain portion of the

code in question is changed.

An effective code search engine allows developers to extract, represent, store, mine and use these source code-specific attributes irrespective of the scale at which all such attributes can expand in size when applied to enterprise or Internet-scale source code repositories.

What's Different in Enterprise Code Search

There are some important differences between enterprise and open-source code search. Open-source code search is done over code repositories found on the Internet and can be seen as an instance of Internet code search—developers searching for code on the Internet. Results can vary widely when searching one's own enterprise code base compared to searching open-source repositories. Inside an enterprise, it's likely there are more stringent code quality checks, better practices for using APIs and stricter code authorship attribution. These are just a few of the factors that can influence the examples developers can find when searching their enterprise code bases.

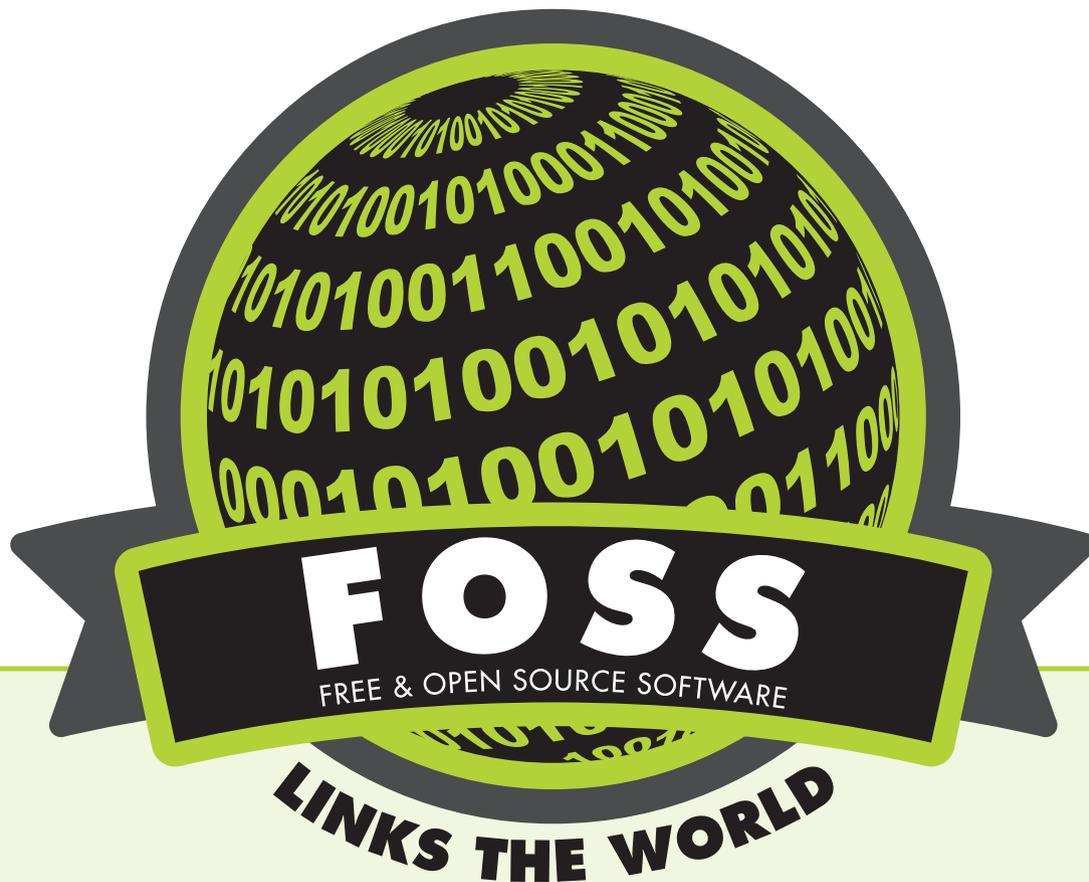
From a tool-builder's perspective, additional benefits of enterprise code search include tighter integration with ALM tools. Tool builders also

can use code search to conduct more accurate analyses during indexing, because code in enterprise source code repositories could be quality controlled or automated to prevent erroneous and incomplete check-ins. In short, there are even more opportunities for us to explore leveraging the unique aspects of enterprise code base.

Measuring the Benefits of Enterprise Code Search

The usage of enterprise code search engines is still in the early adoption phase, so measuring the benefits can be a challenge. Without hard empirical data, these benefits are difficult to quantify but not impossible. Following are examples of how enterprises can assess the benefits:

- As a productivity tool for developers: how much time and effort is spent on questions about code every day? How long does a developer have to wait to get an answer? How much time and effort could a developer save with code search tools, not only herself, but also for other members of the development team who collaborate with her and each other on a daily basis? With a code search tool,



When: September 13, 14 & 15

Where: Greater Columbus Convention Center,
Columbus, Ohio

Keynotes: Robyn Bergeron, Fedora
Marshall Kirk McKusick, FreeBSD
Mark Spencer, Digium
Jon "maddog" Hall

Other Info: OLFI - one day of professional training for system administrators
by companies you know

Saturday Expo

Friday Birds of Feathers sessions

...AND MORE



Ohio LinuxFest 2013
ohiolinux.org

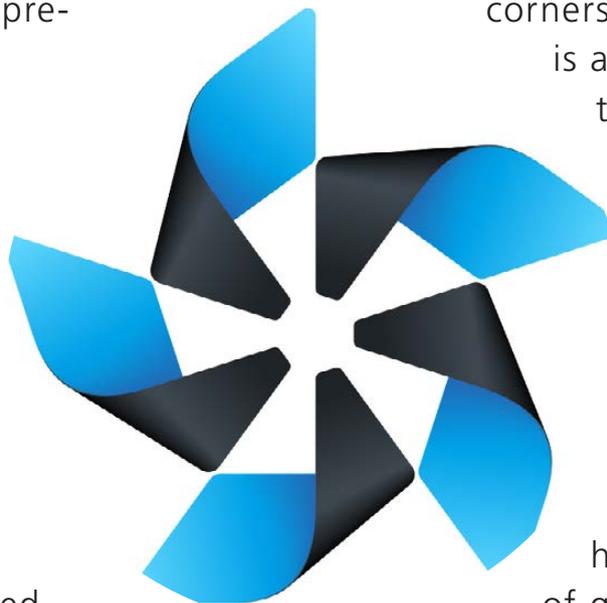
Chances for a Tizen Smartphone Entry

There has been a recent spike in the development of mobile operating systems leading to releases of Firefox OS, Ubuntu for Phones and Jolla Sailfish. Exotic-sounding for sure, but there's one name missing from the list, a mobile-optimized OS that will begin appearing in mass-produced handsets in mid-2013. Tizen is its name, and its developers intend it to power a variety of devices including phones, tablets, vehicles and televisions.

MICHAEL SCHLOH VON BENNEWITZ

Tizen is a fresh new project, but it has roots in several pre-existing platforms including the distributions Moblin, MeeGo and LiMo. According to the Tizen Association, "The mobile marketplace has undergone extensive change over the past few years. New platforms have emerged, new revenue models have been enabled, and innovations

continue to emerge rapidly from all corners of the industry. Tizen is an open-source solution that provides an innovative platform offering a high level of flexibility in service selection and deployment."



Key Players

Tizen's roots and rich history bring a number of groups together and give rise to a management problem. To leverage the competence

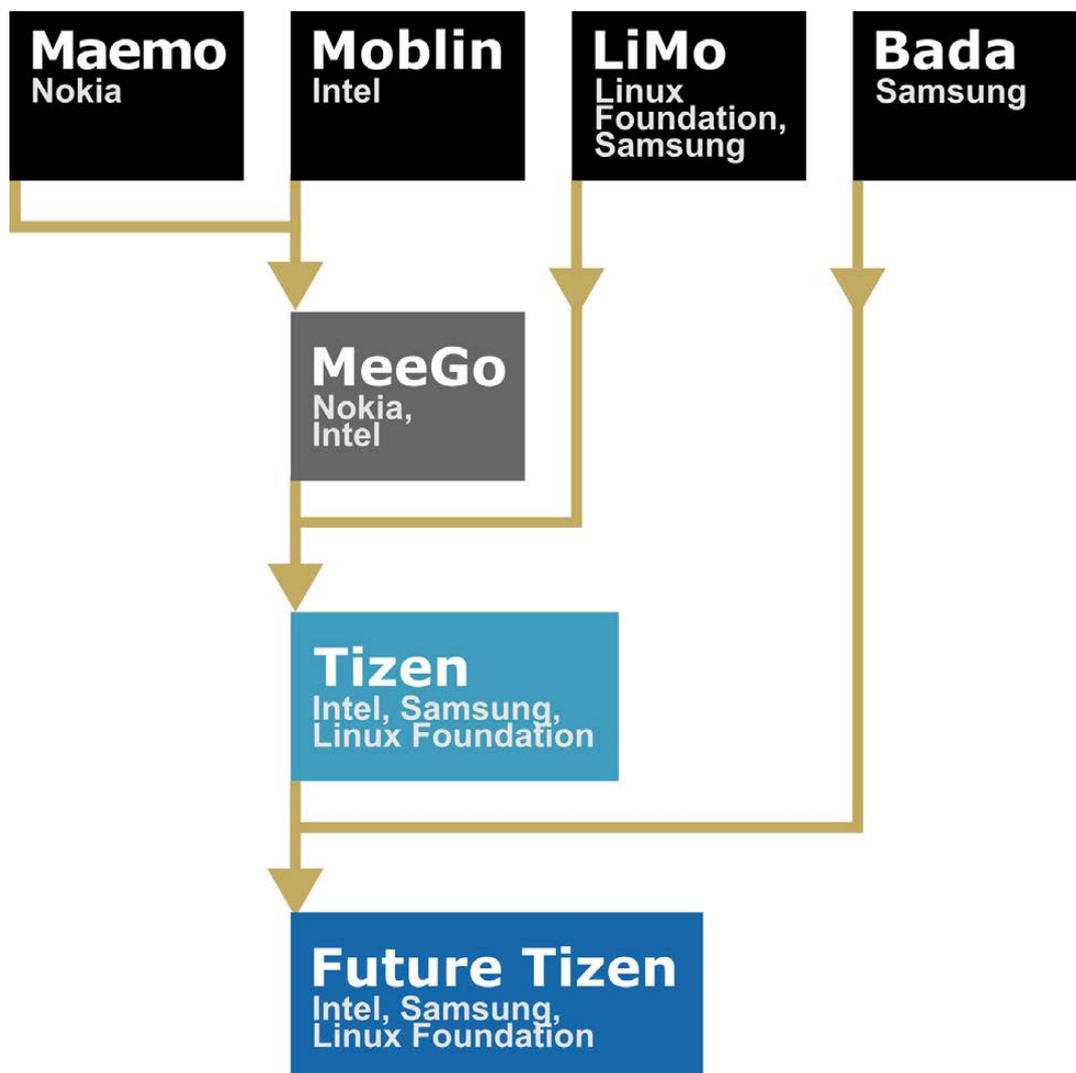


Figure 1. Family Tree of Tizen's Historical Roots (GNU Free Documentation License 1.3)

of the groups involved, Tizen is managed by the following:

- The Linux Foundation: according to the official documentation, the Tizen project “resides” within the Linux Foundation and is governed by a Technical Steering Group.
- The Technical Steering Group: the Technical Steering Group is the primary decision-making body for

Tizen, with a focus on platform development and delivery, along with the formation of working groups to support device verticals.

- The Tizen Association: the Tizen Association has been formed to guide the industry role of Tizen, including gathering of requirements, identification and facilitation of service models, and overall industry marketing and

The Tizen Steering Group directs Tizen's technology while the Tizen Association serves its industrial interests. The Linux Foundation, Intel and Samsung are its largest sponsors.

education. In the Tizen Association's own words, "The Tizen Association's charter is to actively develop the ecosystem around Tizen, which includes the market presence, gathering of requirements, identification and facilitation of service models, and overall industry marketing and education."

- **Corporate Supporters:** the two corporations providing the longest-running support for Tizen are Intel and Samsung. Other manufacturers include NEC, Panasonic, Fujitsu and Huawei. Several telecommunications operator corporations supporting Tizen market adoption include Orange, NTT Docomo, SK Telecom, KT, the Vodafone group, Telefonica and Sprint. Finally, Jaguar Land Rover heads up the automotive market in adopting Tizen for its IVI infrastructure.

- **Huawei Onboard:**

according to analyst David Kerr, VP of Global Wireless Practice at Strategy Analytics, "The addition of Huawei is a significant step forward adding one of the fastest-growing handset vendors in the world and reinforcing the potential for an alternative to Android to develop in both mature and emerging markets for smart devices. The decision by Huawei to support Tizen follows hot on the heels of several other announcements, all of which put pressure primarily on the Android OS and clearly demonstrate that most major vendors and indeed operators continue to hedge their bets as uncertainty and concerns over Google's dominance continue." Sadly, early adopters of Tizen technology have felt little Huawei presence or support so far, the company being suspiciously absent from the recent Tizen developer conference although advertisements stated Huawei sponsorship.

According to one industry insider, the company had planned a conference appearance but backed out of the arrangement, leaving one to hope that other relatively passive players like Fujitsu, Panasonic or NEC are not sitting on the sidelines for lack of optimism.

Dynamics of Intel, Samsung, Google and Motorola

Intel has enjoyed a partnership with Google's Motorola unit, which would seemingly put any future strategic focus on Tizen at odds with Samsung's potential distancing of Android. Additionally, Intel considers Android and Windows to be complimentary technologies. It follows that Intel's market view of Tizen will be similar in the sense that even as Tizen winds up competing with Android for market share, Intel will profit nicely, supporting both platforms with specialized chips and drivers.

In any case, having Intel on board as a Tizen partner will please open-source enthusiasts and Linux users alike. Being the number one commercial contributor to the Linux kernel, "Intel gets the bigger picture of open-source value", says Kaveh Nasri of the Open Source Technology Center. "When you do (open source), the consumer benefits."

Intel Open Source Technology Center

Mr Nasri's group at the Open Source Technology Center is busy developing drivers for the next generation of Intel chips that could soon be powering a Tizen device coming to market, even as Intel competes with several

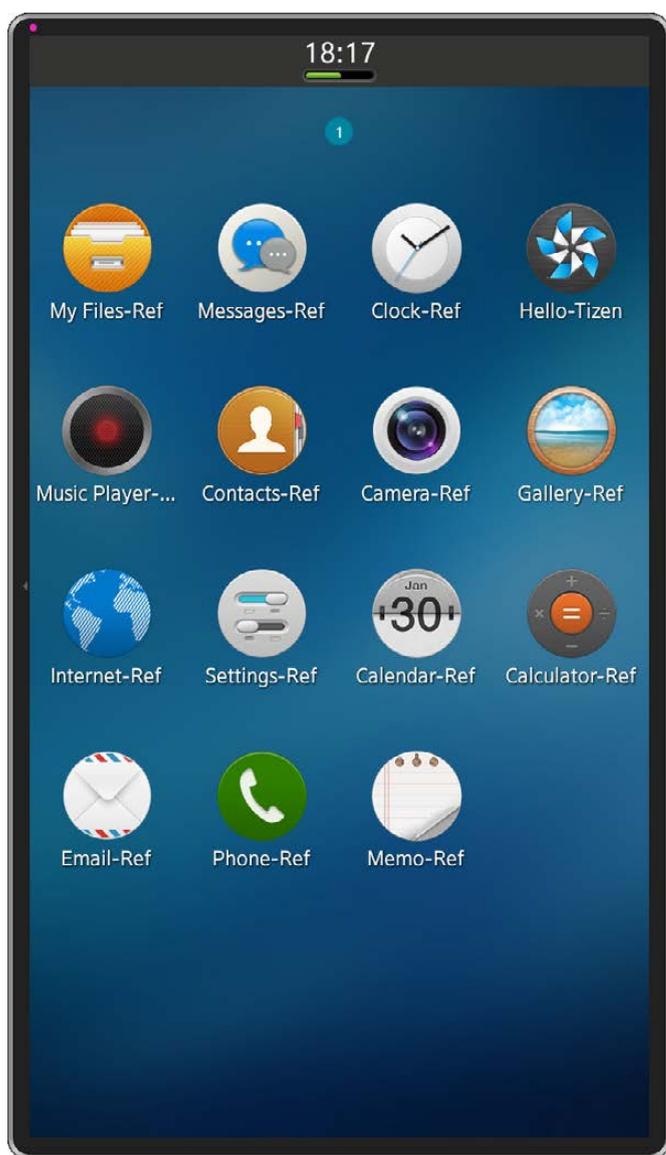


Figure 2. Home Screen of the New Tizen OS (GNU Free Documentation License 1.3)



Figure 3.
The Samsung-Made
Tizen-powered
RD-PQ Handset
Presented at
Mobile World
Congress 2013

other strong mobile semiconductor manufacturers like ARM, NVIDIA and Qualcomm, he reports. Tizen engineers build development images for both IA32 and ARM architectures on a regular basis, and while the scarcely distributed Intel Black Bay handset housed an IA32 Atom

processor, the majority of existing Tizen-powered experimental devices like the Samsung RD-210 and Samsung RD-PQ pack ARM processors.

Lion's Lair of Mobile Competitors

When asked about Tizen's chances as far as competing with established

trend-setters like Android and iOS, midterm arrivals like Blackberry 10 and Windows Phone, or emerging contenders like Firefox OS, Ubuntu Touch or Sailfish, Mr Nasri says that Intel's commercial interests are OS-agnostic. "We just want to sell our silicon", he indicates of Intel's chips and supporting drivers. Linux Foundation Operations Manager Brian Warner provides a different perspective, stating that "Linux collectively does better" when more Linux-based platforms compete. "There is a real opportunity here, and we want to see them all succeed."

Historical Moblin and MeeGo Failures

Such statements are loaded with a rich history of ups and downs at Intel and the Linux Foundation along with associated technologies. Intel is the founder of Moblin and along with the Linux Foundation was a strong supporting partner of MeeGo, the mobile-optimized predecessors to

Tizen. Many would have preferred these technologies to thrive, but their short-lived experimental nature leads some to question Tizen's viability, especially after unraveling corporate partnerships accelerated MeeGo's fall.

Mr Nasri reflects on his 24-year tenure at Intel and remarks, "There's all sorts of alliances and agreements" between Intel and its partners. In the present case, aside from the Linux Foundation, the two key corporate players supporting Tizen are Intel and Samsung. While Samsung likely has its own bet-hedging mobile strategy, Samsung executive VP Jong-Deok Choi gives assurance of full Tizen support in explaining that "Tizen and Android will get along very well."

But a cynical perspective of Tizen's market chances based on Moblin and Atom chips is hard to apply considering Intel's success in the processor races with PowerPC and AMD. Savvy mobile computing enthusiasts may appreciate

Tizen was once almost alone in the category of fledgling mobile operating systems. Now this space is filled by Sailfish, Ubuntu Touch, Firefox OS and others.

Tizen's kernel is Linux and includes a number of familiar userspace tools and libraries. It provides users and developers with a number of freedoms, like application side loading rather than a walled garden.

Intel's long-term contributions to a vibrant open-source ecosystem and protection of computing freedom by supporting community projects like Cordova and Connman. When asked about his alleged opposition to a nearly decided walled-garden deployment approach, chief architect of the Intel Open Source Technology Center Sunil Saxena humbly smiles. His expression as speaking history implies a profound understanding that Tizen's chances of success lie on a solid but open technology foundation.

Early Benefits of Open Source

Mr Nasri agrees that Tizen can beat the odds in contrast to Moblin, MeeGo and even Symbian or Bada. He brings up the important topic of intrusive legal bureaucracy, such as intellectual property constraints typical at large corporations. Open source allows us to do an "end run around the non-disclosure agreement (NDA) lawyers", he states.

The Flora License

Just one problem exists with the open-source approach taken by Tizen. Original Tizen source code (as opposed to integrated components) is licensed under terms of the Flora license, which is not approved by the Open Source Initiative (OSI). In theory, this failure to obtain OSI's blessing places Tizen somewhere in between proprietary and open-source platforms. In practice, this choice could affect a number of interested parties. Assuming that many would otherwise use Tizen for its added development and operating freedom, the choice of a Flora license is unfortunate and may cloud management's ability to judge Tizen's legal viability in their markets.

Attracting Developers

An industry insider familiar with intellectual property (IP) law states, "likely at this stage our only real allies are the open dev crowd, those who like Tizen because it is open and a real Linux." The insider claims that, "Firefox



Figure 4. GUI Tools Distributed in the Tizen SDK 2.1 (GNU Free Documentation License 1.3)

OS dev phones sold out almost instantly” for just this reason.

Bada Overreach

Nevertheless, cracks may be appearing in Tizen’s developer tech armor due to decisions relating to overreach of a scheduled Bada migration. An industry insider refuses cooperation in this area, saying instead “I am not going to be the guy who is asked why is Bada used now instead of EFL?”, clearly fearing that Tizen’s pristine and future-proof architecture may be infected with legacy Bada logic.

Market Trend Analysis

High-ranking corporate policy-makers happily state their optimism of the approaching Tizen smartphone market entry. Samsung executive VP

Jong-Deok Choi expresses his assurance that, “We have very high expectations”, and “Tizen is very real.” But what strategy is behind the policy of strong industrial support for Tizen’s rollout? To understand the various corporate strategies involved in shaping Tizen’s future, as well as obtaining a realistic interpretation of Tizen’s environmental market trends, informed opinions by diverse analysts give added value.

We ask if Samsung could be using Tizen as a hedge against Android’s ever-growing market dominance, leading OEMs and chip-makers to play cat and mouse in adapting to the fragmented and controlling Android platform. Analysts do speculate along these lines that since Google acquired Motorola Mobility, Samsung and other smartphone

The planned Bada migration to Tizen is a recent policy development and is dynamic in nature. Some fear a future overreach when integrating Bada logic.

Analysts at Gartner, IDC and Strategy Analytics are keeping a watchful eye on Tizen and its competitors. Some forecast an increased market share at Android's expense.

manufacturers would be seeking alternatives to Google's Android. Android could become the preferred platform for Motorola, which likely would put competing manufacturers at a disadvantage.

Gartner Figures

Gartner's latest figures put Android's market share at nearly 75% of world-wide smartphone adoption. "With new OSes coming to market, such as Tizen, Firefox and Jolla, we expect some market share to be eroded but not enough to question Android's volume leadership", states Gartner principal research analyst Anshul Gupta.

IDC Research

Indeed, as Android adoption has shot sky-high, a number of new mobile platform contenders have lined up to compete. IDC mobile-phone research manager Ramon Llamas and IDC worldwide quarterly mobile-phone tracker senior research analyst Kevin Restivo remark that "This is shaping up to be a

pivotal year for the open-source operating system, as multiple platforms, including Mozilla, SailFish, Tizen and Ubuntu are expected to introduce or launch their first smartphones in the coming months."

Strategy Analytics

Strategy Analytics analyst Scott Bicheno continues, "Android will remain the dominant smartphone OS for the next few years, but its share will peak in 2013, while the global share of iOS is unlikely to grow much further so long as Apple maintains its premium-tier-only strategy. While Microsoft's market share will remain small in 2013, it will emerge as the clear third-placed platform by 2017, with BlackBerry leading a chasing pack that will include the nascent smartphone platforms: Tizen, Firefox OS, Ubuntu and Sailfish."

Other Opinions

But while Samsung has profited nicely from its widespread Android integrations, some analysts

have criticized it for failing to innovate. Using Google's software is leading to a risky dependence on the software giant's technology even after its strategic acquisition of Motorola's mobile division, a direct competitor to Samsung. They reason that Tizen enjoys additional support due to these market developments. "Samsung has grown up and is playing as the big guy, moving away from Android", says Gartner Mobile Devices Research VP analyst Carolina Milanesi.

Kevin Burden, an analyst at Strategy Analytics, agrees that Samsung is motivated to distance itself from Android for this reason but also partly because it wants greater control over the operating system in its phones. "It almost feels like Samsung is trying to set up Tizen as its next OS instead of Android", he says.

Operator and Service Provider Tactics

Vice President of Yandex Labs Juggs Ravalia puts forth the theory that many corporate heavyweights

(especially operators) in the Tizen Association are worried about lock down in Android APIs and want to free their technology from difficulties circumventing Google-dictated services like Maps. While standing to benefit strategically from both added development freedom and a Tizen success, operators like NTT Docomo have confirmed that Tizen is a very operator-friendly platform.

Orange (France Telecom)

This bodes well for the multinational operator Orange. According to Frédéric Dufal, Technical Director of Orange Devices and Vice Chairperson of the Tizen Association, the launch of Orange's first Tizen device will occur in select European markets at the end of summer 2013 with more devices coming at later launch dates.

But Orange is not just interested in the traditional smartphone markets, rather "I hope Tizen someday will be delivering great stuff for emerging markets and especially Africa, and

Orange has announced a forthcoming Tizen handset offering. Its end of summer rollout will include popular Orange services leveraging Tizen's superb HTML5 support.

Tizen managers and developers alike are working to provide a variety of game engines to early adopters with hopes of offering games from industry giants like Unity.

Orange is very active there”, says Mr Dufal.

In fact, although Orange and France Telecom have strong commercial representation in many African countries due to historical French trade relations, they share a second commercial interest in meeting demand for less-expensive smartphone technology served by competing multinational operators like Telefonica with their Firefox OS handsets. Should either or both such efforts succeed, then mobile users in emerging markets win in the end, but what does this mean for Tizen? Mr Dufal answers, “We hope that in the longer term, we can use Tizen to democratize the access to the Internet and to smartphones in emerging markets especially in Africa and the Middle East, where not everybody can afford the fancy smartphones.” His words could resonate with users in expense-reduced hardware markets producing the likes of OLPCs and Raspberry Pis.

Yandex and Big Data

Internet companies like Yandex

are turning to Tizen for its help in solving problems with and improving cutting-edge developments like navigation, assisted transportation searches and in vehicle infotainment (IVI.) According to Vice President of Yandex Labs Juggs Ravalia, Tizen could serve as a platform for a new generation of services only possible through the innovative combination of big data and free software APIs like those provided by Tizen.

Automobile Industry and IVI

It’s no surprise that Yandex is working on transportation-relevant network services. According to Senior Technical Specialist for infotainment systems at Jaguar Land Rover Matt Jones, “uptake of HTML5 in vehicles is going through the roof”, and that “Linux is running in over a million vehicles already.” Mr Jones goes on to state the utility and consumer thirst for rich IVI systems, for which Tizen is a perfect platform match.

Available Game Engines

Everybody knows that

smartphone consumers are enchanted by platforms stocked with eye-catching games. A number of companies providing game platforms, engines and more than a dozen game technologies have come forward to meet Tizen's game needs, including big names like Havok, Unity 3D, Yoyo Games, Marmalade and Gamesalad. Tizen's potential as a game platform is attracting independent developers of HTML5 technology as well, with companies like Sencha Touch providing JavaScript abstraction

libraries for accelerated development.

Primary Game Hardware and Market Size

Speaking on behalf of one of the largest game engines, Unity General Manager John Goodale emphasizes ubiquity when speaking of a giant mobile market large enough to accommodate a number of new technologies. The smartphone market supports an "explosive industry that is growing very rapidly", states Mr Goodale. "There's not just room for

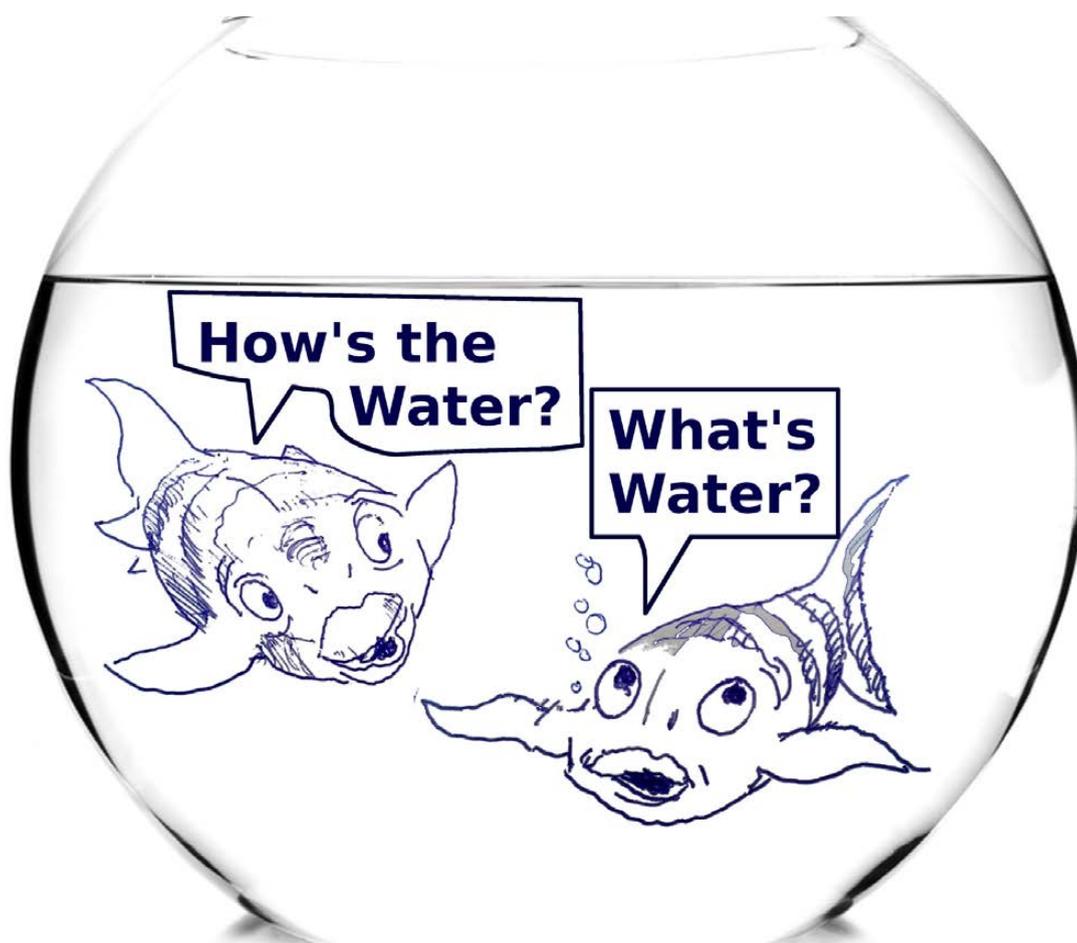


Figure 5. Some things are so commonplace that we hardly notice them. (GNU Free Documentation License 1.3)

Tizen may raise a few eyebrows. Aside from being packed with familiar Linux technology, Tizen sports some unique features like dynamic boxes and hybrid application packages.

two or three or a handful of players”, rather “the market can extend to as far as we can effectively execute”, he says. Regarding the ubiquitous nature of mobile technology creeping into our lives over time, Mr Goodale continues, “some things have become so commonplace that we hardly notice them”, and that according to Juniper Research, mobile devices like Tizen smartphones likely will be the primary hardware for gaming by 2016.

Referring to Unity’s decision to support Tizen (as well as Tizen’s entry in the smartphone market) Mr Goodale summarizes, “Jump on in, the water’s warm.”

Uninspiring Ubiquity or Technical Distinction

A number of technical characteristics set Tizen apart from other systems with similar mobile-oriented goals. Aside from any number of eye-candy features likely to be implemented close to a first device launch date, Tizen designers will need to strive for unique features to

secure technical distinction that sets Tizen apart from its competition. Consumers interested in such unique technology include a number of actors along the technical “food chain” starting with designers to programmers and finally end users.

The Filesystem

For Linux users, first and foremost is the familiar layout of Tizen’s internal filesystem. Most configuration can be found in /etc, runtime variable state in /var, user files in /home/<user>, temporary files in /tmp and so on. While security abstraction measures exist to mark and protect certain regions (SMACK), this filesystem familiarity will surely provide comfort to some.

Dynamic Boxes

Tizen puts forth the concept of dynamic boxes, small Web applications embedded inside other applications, to provide users with dynamically updated content. The rich Tizen API exposed to provide dynamic box logic supports the dynamic box with an independent life

cycle. At runtime, Tizen's Web runtime has the ability to control the life cycle of dynamic boxes.

Ownership and Other End-User Freedoms

Compared with nearly all existing mobile platforms, Tizen offers an unrivaled degree of end-user freedom. A user can modify or replace any part of the platform right down to the kernel and low-level security layers. Rather than blurring the lines of free license by releasing binary blobs of kernel and libc while publishing only sanitizing header files, Tizen's GNU/Linux kernel and other sources are complete, on-line and publicly

accessible. Developers can pull a copy of these sources and build their own Tizen image ready for installation to hardware. It remains to be seen if operators will implement tricky bootloaders to lock terminals to custom kernels and certain Tizen drivers depending on proprietary microcode (like the modem providing cellular voice communication), but as far as platforms go, Tizen provides the end user with far-reaching freedoms.

Breadth of Supporting Architecture

Arguably, from a development perspective, Tizen's unique platform architecture sets it apart from nearly

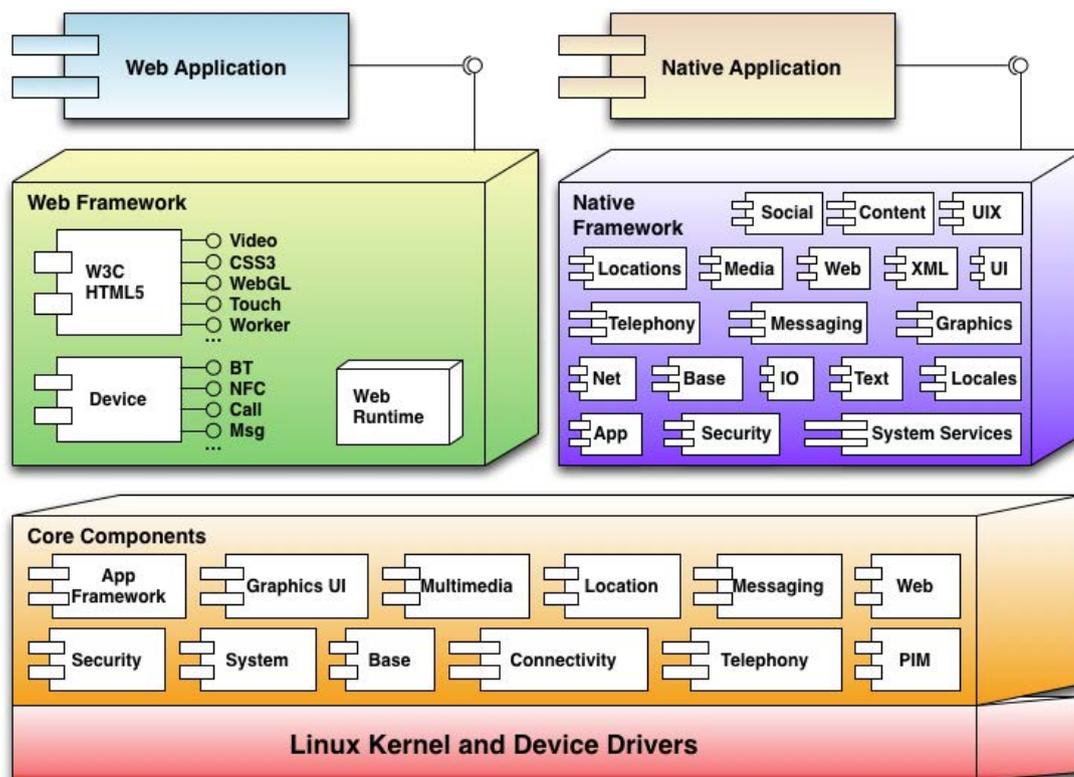


Figure 6. Architecture of the Tizen SDK 2.1 (CC Attribution 3.0 Unported)

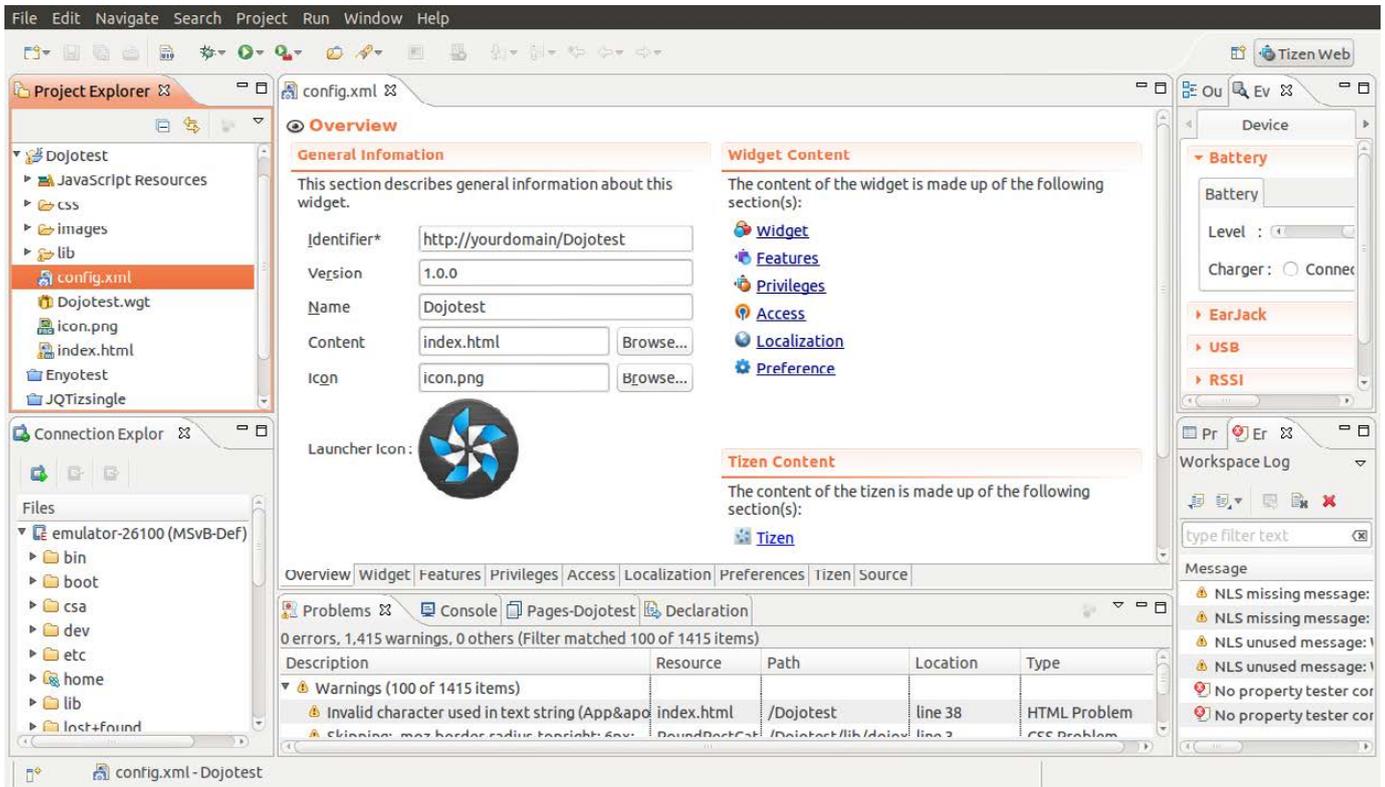


Figure 7. The Tizen SDK's Eclipse-Based IDE (GNU Free Documentation License 1.3)

all competitors with the exception of BlackBerry. Tizen's architects eyed a variety of device types from the beginning, leading to a flexible architecture that will accommodate all sorts of tablets, desktops, vehicle terminals (IVI), television consoles and others once the first wave of smartphone handsets is rolled out. Furthermore, Tizen's layered architecture features core components and frameworks providing APIs to high-level applications of a variety of technologies. This breadth of logic will appeal to developers of Web, native, hybrid and third-party technologies alike.

Web Framework Support

Scoring 492 of 500 points at html5test.com, "Tizen support for HTML5 is the best among mobile browsers", says Samsung executive VP Jong-Deok Choi. Speaking of its Web runtime and Web framework support, "Tizen is the platform most compliant with HTML5 standards", agrees Mr Dufal. It's also important to note that security features like content security policy (CSP) are built in to Tizen's Web runtime as well.

Regarding the question of how high Tizen will rise by leveraging Web technologies, W3C mobile Web initiative activity lead Dominique Hazael-Massieux replies, "Tizen is very

well positioned for sure.”

Native Framework Support

Developers accustomed to POSIX server technology can port their existing IA32 server software quickly and easily using the tools freely available in the Tizen SDK (supporting Linux, OS X and Win32 OS types). Packaging the ported software to RPM files and installing to the device is straightforward. Other developers of client apps can choose from Web or native

frameworks and deploy from the Tizen Store in the usual way.

Integral Hybrid Packaging Support

Tizen includes logic to run native (often server) code alongside Web (often client) code packaged together, providing a convenient transport for complex applications requiring device-specific features (for example, an SSL crypto library) as well as a high-level UI (for example, Web-based for ease of maintenance).

Providing a special native application framework, Tizen supports POSIX development as well as full OpenGL ES hardware accelerated graphics development using the Enlightenment Foundation Library (EFL).

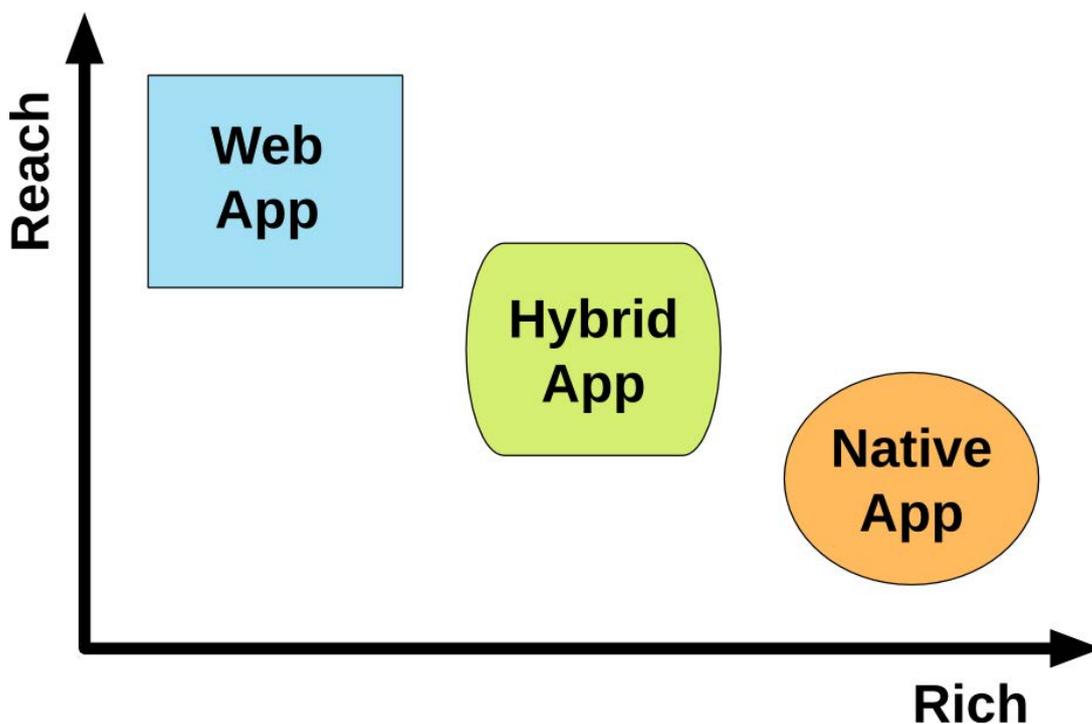


Figure 8. Tizen supports a broad range of technologies in mobile apps (GNU Free Documentation License 1.3).

Engineers of portable Tizen applications have the choice of both Cordova (Phonegap) and Appcelerator Titanium Studio for JavaScript-based third-party abstraction.

Third-Party Hybrid Abstraction Support

Finally, third-party providers of system abstraction frameworks like Adobe Phonegap, Apache Cordova and Appcelerator Titanium serve to fill any technical API gaps and facilitate porting of existing applications even further.

Hope for the Less-Savvy End User

It remains to be seen how enthusiastic a less tech-savvy user will be about such distinctive technical features, but such users may indirectly profit from quick porting of outstanding applications in other mobile OS distributions. They could benefit additionally from high-powered compiled applications running on Tizen's native APIs when such an architecture is relevant. This model appears to match Research in Motion's efforts with its Blackberry 10 release; however, closer inspection reveals nuances in handling of Web applications by the respective Web runtimes as well as obvious differences in graphics widget toolkits and POSIX implementations.

Application Deployment and the Tizen Store

According to the Tizen Association, "As part of the Tizen Association's focus on ecosystem development, the Tizen Store will launch later in 2013 with thousands of apps, allowing developers to monetize their work and creating a robust ecosystem. The outreach to app developers to build HTML5 apps has begun."

It remains to be seen how favorably developers will take to Tizen's store or how enthusiastically consumers will use it. Although Director of Systems Engineering Mark Skarpness emphasizes that "The Tizen App Store is open for business", neither APIs nor client applications have been revealed. Nevertheless, developers already are free to open accounts and submit applications for no charge. Even better, sponsors with deep pockets have announced official Tizen developer contests awarding impressive prizes. According to the Tizen Association's planned Tizen App Challenge, "With over \$4M in cash prizes, there's never been a better time to create or

Resources

Tizen Vision: <http://www.tizenassociation.org/vision>

Tizen Members: <http://www.tizenassociation.org/members>

About Tizen: <http://www.tizen.org/about>

“Can Tizen Challenge Android with Huawei Now Onboard?”:
<http://blogs.strategyanalytics.com/WSS/post/2012/02/29/Can-Tizen-challenge-Android-with-Huawei-now-onboard.aspx>

“Gartner Says Asia/Pacific Led Worldwide Mobile Phone Sales to Growth in First Quarter of 2013”: <http://www.gartner.com/newsroom/id/2482816>

“Android and iOS Combine for 92.3% of All Smartphone Operating System Shipments in the First Quarter While Windows Phone Leapfrogs BlackBerry, According to IDC”:
<http://www.idc.com/getdoc.jsp?containerId=prUS24108913>

“Global Smartphone Sales Forecast by OS for 88 Countries and 14 Operating Systems: 2007 to 2017”: <http://sa-link.cc/WSS240513>

“Samsung offers barely a mention of Android amid Galaxy S4 hoopla”:
http://www.computerworld.com/s/article/9237618/Samsung_offers_barely_a_mention_of_Android_amid_Galaxy_S4_hoopla

TDC13—Keynotes: Thursday May 23, 10:45: http://www.youtube.com/watch?feature=player_embedded&v=Ddv_OrbMTyg

Tizen Game Development: http://wiki.tizen.org/wiki/Game_development

Tizen 2.1 Release Notes: <http://developer.tizen.org/downloads/sdk/2.1-release-notes>

“The Definitive Guide to Developing Portable Tizen Apps”: <http://mobile.dzone.com/articles/definitive-guide-developing>

“The opportunity of HTML5 and TIZEN”, Frédéric Dufal: http://cdn.download.tizen.org/misc/media/conference2013/slides/TDC2013-The_Opportunity_of_HTML5_and_Tizen.pdf

Tizen Association Celebrates Progress and Discusses the Future:
<http://www.tizenassociation.org/tizen-association-celebrates-progress-and-discusses-future>

Tizen App Challenge: <http://developer.tizen.org/contests/tizen-app-challenge>



LINUXCON
NORTH AMERICA 2013



cloudopen
NORTH AMERICA 2013

SEPTEMBER 16-18, 2013 · HYATT NEW ORLEANS

FEATURING KEYNOTE SPEAKERS

Jonathan Bryce

Executive Director at OpenStack

Candy Chang

TED Senior Fellow & Artist

Chris DiBona

Director of Open Source at Google

Dirk Hohndel

Chief Linux & Open Source Technologist at Intel

Kevin Kelly

Senior Maverick at Wired

Gabe Newell

Co-Founder, Valve Corporation

Linus Torvalds

Linux Creator and Linux Foundation Fellow

Eben Upton

Founder, Raspberry Pi Foundation

Jim Zemlin

Executive Director at The Linux Foundation

USE DISCOUNT CODE **LCPRM20** FOR A
20% DISCOUNT TO ATTEND BOTH EVENTS
REGISTER NOW AT
EVENTS.LINUXFOUNDATION.ORG

 **LINUX FOUNDATION**



DOC SEARLS

Dear Hotels: Quit Being A-holes

Sphinctered connectivity on the pay toilet model makes a lie of the term “hospitality”. It’s also a working model for the mobile Internet—and that’s the main issue.

Bob Frankston says connectivity will eventually become “ambient”—something we just assume, much as we assume electricity, water, sewage treatment and other infrastructural conveniences. None of those conveniences are free of cost, of course, and we pay for them one way or another. As utilities, it is normal for those paying for them to share reasonable use of them for free with others. Thus, we assume that, for example, a restroom in a hotel or gas station has a sink with running water, a light that goes on and a toilet that flushes. In less-developed parts of the world, or away from those conveniences, we make do with less, or on our own.

But civilization requires that certain conveniences are available as a matter of course and are offered by those who pay for them directly as a simple grace to others.

This is not yet the case with Internet connectivity, especially in the “hospitality” industry.

I am facing this fact at the Novotel Lakeside (<http://www.novotel.com/gb/hotel-5308-novotel-queentown-lakeside/index.shtml>), an otherwise fine hotel in Queenstown, New Zealand. Here my Internet connection is so sphinctered that all I can do is contemplate the problem at hand rather than the original subject of this month’s column. I cannot continue writing about that subject because to do so would require that

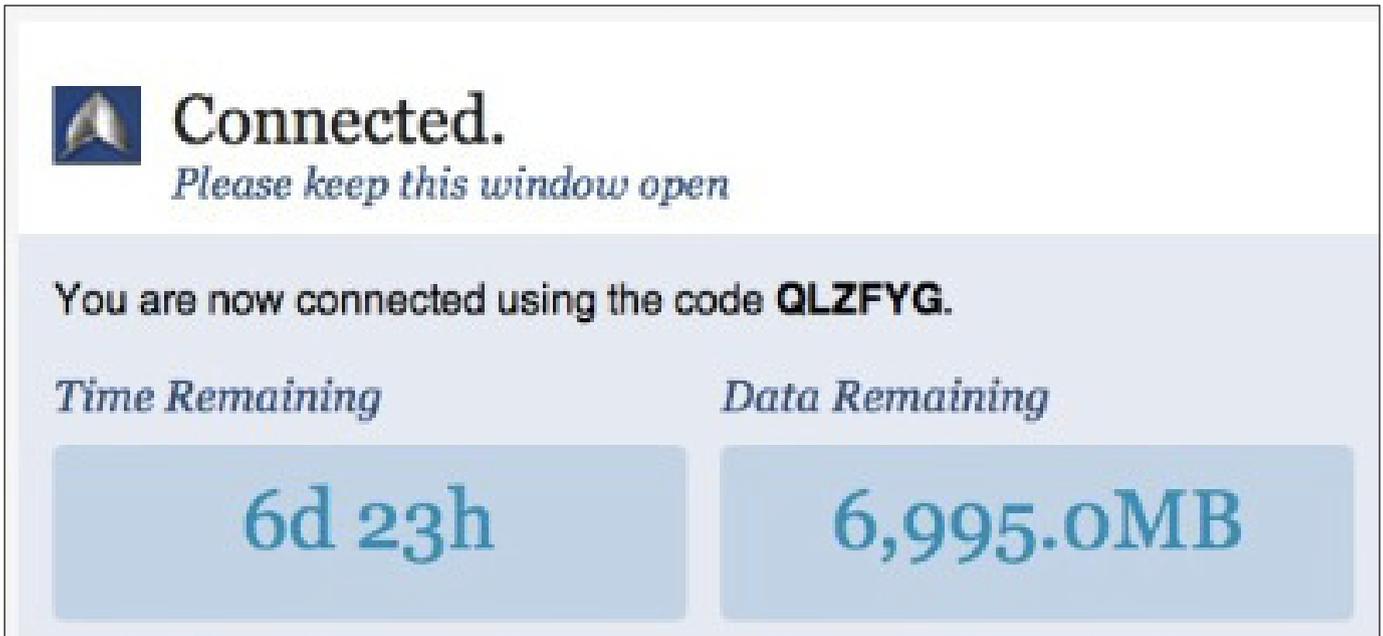


Figure 1. Screenshot of Doc's Time and Data Remaining

I use the Internet in a fully interactive way. What I have instead is a connection that has suddenly slowed to a tortuous crawl. This happened after the hotel cut me off and then offered to let me proceed at a slow pace or pay \$.10/MB (or about \$100/GB) for the full-speed connection I thought I would have for the 7GB that already cost me \$115 at the start of my stay.

By that deal, I had seven days to use the 7GB, and up to four devices I could connect in my room, over Wi-Fi. I thought it was worth paying for, even though we were staying only for three days, and it was unlikely that we'd use

7GB of data. It also was the most expensive deal offered, so I thought it would cover the most use, with the most convenience. Instead, it was less a bait-and-switch than a bait-and-whack.

To get a sense of my frustration at the moment, consider what I am looking at right now on my screen (Figure 1).

The problem with his message is that 6,995.0MB is *not* what's remaining. That's how much I might now pay \$.10/MB for, or \$699.50 if I eat through the whole thing.

So, in my frustration and confusion, I just went down to the front desk, where they printed

Welcome to Novotel Queenstown Lakeside Internet Access System
For assistance call our 24-hour Internet helpdesk: 0800 116 113

Select a plan »

2-hours \$14.90, 100MB	24-hours \$29.90, 300MB	7-days \$115.00, 1GB
----------------------------------	-----------------------------------	--------------------------------

7-day plan

\$0.68 per minute up to \$115.00 for 7 days from when you first connect. After you have reached 1000MB, you will have the choice of paying \$0.10 per MB to transfer more data at maximum speed or having your connection speed slowed without incurring additional charges.

For example, if you start your enrolment at 1PM today then it will finish at 1PM on Fri 28th June 2013 or when you transfer more than 7,000MB of data.

 *This plan includes the ability to connect up to 4 simultaneous devices.*

Figure 2. Screenshot from Doc's Hotel

out a more readable form of this (Figure 2).

(Note: I copied the screenshot shown in Figure 2 and inserted it later, when I had a better connection.)

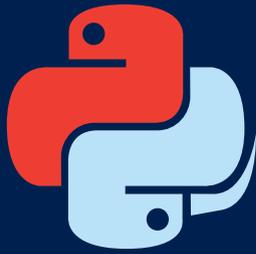
Never mind the insanity of torturing customers with this strange mix of conditionalities. The market will fix that stuff eventually. (And I'll do my part with this column.) Think instead of negative vs. positive economic externalities.

On the negative side is the unlikelihood that I will ever stay in this hotel again—or in any other Accor Hotel (<http://www.accorhotels.com/gb/usa/index.shtml>), all of which, I gather, have the same aversive Internet offering. Also on the negative side, for the likes of Accor, is my

preference these days to stay in AirBnB homes, for the simple reason that *all* of the ones I consider have good Internet connections, and *none* of them see their Internet connection as the digital equivalent of a pay toilet.

On the positive side, think about how Linux—and everything developed by geographically separated creators over the Internet—requires easily available and low-cost connections—and which then in turn produce even more products and services with positive economic externalities.

The main problem is that we're dealing with a new and awful norm here: metering the Internet as if it were an old-fashioned phone service.



KIWI PYCON
AUCKLAND 2013
THE YEAR OF THE SNAKE

AUT Sir Paul Reeves
Building City Campus

SEPTEMBER
6-8



New Zealand's premier
Python event of the year
<http://nz.pycon.org/>



Confusopolies are very complicated shell games in which no customer can intuit, much less find, a first cost.

Although not verbatim, both the hotel and the help desk on the phone told me “all the hotels work this way”. It could be that that’s true in New Zealand, although I doubt it. In the US and Europe, the expensive hotels are the ones with inconvenient connectivity deals (although I’ve seen none with data caps or metered usage). It’s the cheap hotels that offer free Internet, just like they offer free electricity, heat, air conditioning and running water.

In the wired parts of the Internet, where we connect by Ethernet through fiber, cable TV or phone lines, we tend not to sense prices for sums of data, even if there are “caps” involved. Comcast, for example, has “flexible” terms surrounding its 250GB/month data caps (<http://customer.comcast.com/help-and-support/internet/common-questions-excessive-use>). But in the wireless parts of the Net connected over 3G and 4G/LTE connections, the “caps” are very present and constantly threatening. They are also much lower

than we see with wired, and with costs that are much higher. (See, for example, AT&T’s and Verizon’s plans: <http://www.att.com/shop/wireless/data-plans.html#fbid=LpfyALywhZw> and <http://www.verizonwireless.com/wcms/consumer/shop/shop-data-plans.html>.)

If we follow the model set by expensive hotels and mobile phone companies, the Net will turn into a complicated “service” on the model of phone and cable systems, rather than the much simpler model of pure utilities with boundless positive economic and social externalities, such as we have with electricity, water and sewage treatment. This is a huge fork in the road of the Net’s future.

Back when he was Chief Scientist at BT, JP Rangaswami said the core competence of phone companies was not communications, but billing. As a group they are very successful at that. The result is what Scott Adams calls a “confusopoly” (<http://www.att.com/shop/wireless/data-plans.html#fbid=LpfyALywhZw>):



10th Annual
**HIGH PERFORMANCE COMPUTING
 FOR WALL STREET** Show and Conference

SEPTEMBER 9, 2013 (MONDAY) ROOSEVELT HOTEL, NYC

Madison Ave and 45th St, next to Grand Central Station

**SAVE
 THE DATE**

**Big Data, Cloud, Linux, Low Latency, Networks, Data Centers, Cost Savings.
 Wall Street markets will assemble at the 2013 HPC Sept. 9.
 See new systems live on the show floor.**

This 10th Annual HPC networking opportunity will assemble 800 Wall Street IT professionals at one time and one place in New York in September 2013.

This HPC for Wall Street conference is focused on **Speed, Low Latency, Networks, Data Centers, lower computer costs.**

Our Show is an efficient one-day showcase and networking opportunity.

Leading companies will be showing their newest live on-the-show floor.

Register in advance for the full conference program which includes general sessions, drill down sessions, an industry luncheon, coffee breaks, exclusive viewing times in the exhibits, and more. Save \$100. \$295 in advance. \$395 on site.

Don't have time for the full Conference? Attend the free Show.

Register in advance at: www.flaggmt.com/hpc



Wall Street IT speakers and Gold Sponsors will lead drill-down sessions in the Grand Ballroom program.



Show Hours: Mon, Sept 9 8:00 - 4:00
 Conference Hours: 8:30 - 4:50

September 2012 Sponsors



Show & Conference: Flagg Management Inc
 353 Lexington Avenue, New York 10016
 (212) 286 0333 fax: (212) 286 0086
flaggmt@msn.com

www.flaggmt.com/hpc