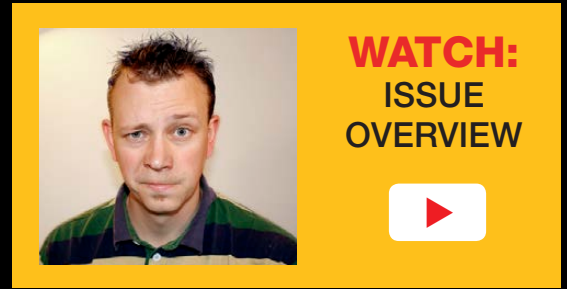


Automate Application Deployment with MCollective

# LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community



JANUARY 2017 | ISSUE 273  
<http://www.linuxjournal.com>

# Password Managers and Online Security

PASSWORD

Data Formats  
and Protocols  
for the **IoT**

A Look at  
the **Synology**  
**NAS Server**

Analyze  
Videos with  
**Tracker**

**Practical books  
for the most technical  
people on the planet.**

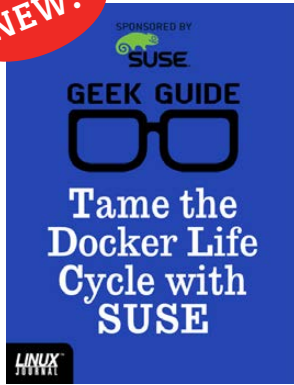
# **GEEK GUIDES**



**Download books for free with a  
simple one-time registration.**

**<http://geekguide.linuxjournal.com>**

NEW!



## Tame the Docker Life Cycle with SUSE

**Author:** John S. Tonello  
**Sponsor:** SUSE



## SUSE Enterprise Storage 4

**Author:** Ted Schmidt  
**Sponsor:** SUSE



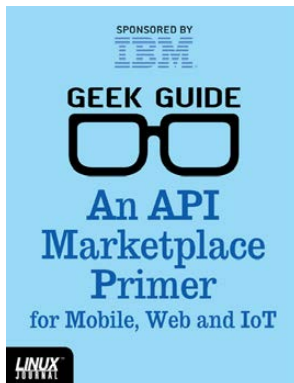
## BotFactory: Automating the End of Cloud Sprawl

**Author:** John S. Tonello  
**Sponsor:** BotFactory.io



## Containers 101

**Author:** Sol Lederman  
**Sponsor:** Puppet



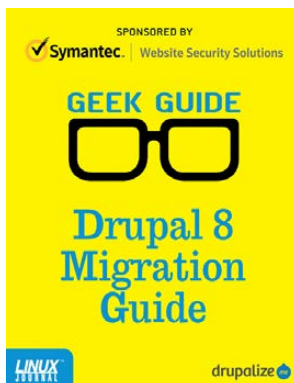
## An API Marketplace Primer for Mobile, Web and IoT

**Author:** Ted Schmidt  
**Sponsor:** IBM



## Public Cloud Scalability for Enterprise Applications

**Author:** Petros Koutoupis  
**Sponsor:** SUSE



## Drupal 8 Migration Guide

**Author:** Drupalize.me  
**Sponsor:** Symantec



## Beyond Cron, Part II: Deploying a Modern Scheduling Alternative

**Author:** Mike Diehl  
**Sponsor:** Skybot

# CONTENTS

JANUARY 2017  
ISSUE 273

## FEATURES

### 76 Online Privacy and Security Using a Password Manager

Take charge of your online account credentials.

**der.hans**

### 92 Low Power Wireless: CoAP

A look at the application layer: device description using CoRE Link Format, data formats using CBOR and REST APIs.

**Jan Newmarch**

PASSWORD

# COLUMNS

**38** Reuven M. Lerner's  
**At the Forge**

Testing Models

**46** Dave Taylor's  
**Work the Shell**

Is the Moon Waxing or Waning?

**52** Kyle Rankin's  
**Hack and /**

Orchestration with MCollective,  
Part II

**58** Shawn Powers'  
**The Open-Source  
Classroom**

My Love Affair with Synology

**106** Doc Searls' EOF

Debugging Democracy

# IN EVERY ISSUE

**8** Current\_Issue.tar.gz

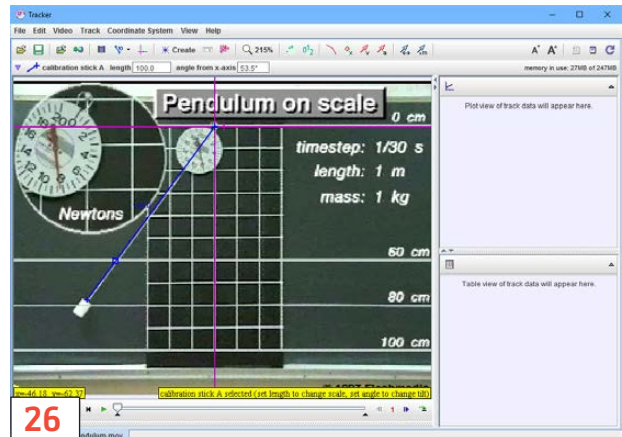
**10** Letters

**16** UPFRONT

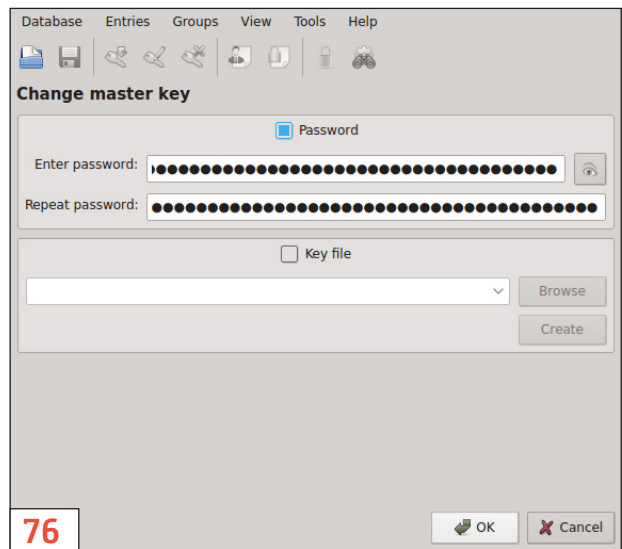
**36** Editors' Choice

**68** New Products

**114** Advertisers Index



**58**



**76**

### ON THE COVER

- Automate Application Deployment with MCollective, p. 52
- Password Managers and Online Security, p. 76
- Data Formats and Protocols for the IoT, p. 92
- A Look at the Synology NAS Server, p. 58
- Analyze Videos with Tracker, p. 26

# 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

<b>Executive Editor</b>	Jill Franklin jill@linuxjournal.com
<b>Senior Editor</b>	Doc Searls doc@linuxjournal.com
<b>Associate Editor</b>	Shawn Powers shawn@linuxjournal.com
<b>Art Director</b>	Garrick Antikajian garrick@linuxjournal.com
<b>Products Editor</b>	James Gray newproducts@linuxjournal.com
<b>Editor Emeritus</b>	Don Marti dmarti@linuxjournal.com
<b>Technical Editor</b>	Michael Baxter mab@cruzio.com
<b>Senior Columnist</b>	Reuven Lerner reuven@lerner.co.il
<b>Security Editor</b>	Mick Bauer mick@visi.com
<b>Hack Editor</b>	Kyle Rankin lj@greenfly.net
<b>Virtual Editor</b>	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

---

**President** Carlie Fairchild  
publisher@linuxjournal.com

**Publisher** Mark Irgang  
mark@linuxjournal.com

**Associate Publisher** John Grogan  
john@linuxjournal.com

**Director of Digital Experience** 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

Nick Baronian  
Kalyana Krishna Chadalavada  
Brian Conner • Keir Davis  
Michael Eager • Victor Gregorio  
David A. Lane • Steve Marquez  
Dave McAllister • Thomas Quinlan  
Chris D. Stark • Patrick Swartz

## 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.

**STORAGE  
REDEFINED:**

**You  
cannot  
keep up  
with data  
explosion.**

Manage data expansion with SUSE Enterprise Storage.

SUSE Enterprise Storage, the leading open source storage solution, is highly scalable and resilient, enabling high-end functionality at a fraction of the cost.

[suse.com/storage](https://suse.com/storage)



# Insecure about Being Unsecure



**SHAWN  
POWERS**

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 email at [shawn@linuxjournal.com](mailto:shawn@linuxjournal.com). Or, swing by the [#linuxjournal](#) IRC channel on [Freenode.net](http://freenode.net).

NEXT  
Letters



There have been epic battles over whether “insecure” or “unsecure” should be used when referring to computer security. Granted, those epic battles usually take place in really nerdy forums, but still, one sounds funny and the other seems to personify computers. Whichever grammatical construct you choose, the need for security is greater now than ever. As Linux users, we need to make sure we're not overconfident in the inherent security of our systems. Remember, they all have a weak link: us.

Reuven M. Lerner starts off this issue by showing how to test the machine learning model he created last month. If computers are only as good as their programming, it's important to test how well the code learns. Sometimes coming up with a good test is harder than writing the original code!

Dave Taylor finishes his werewolf-warning



**VIDEO:**  
Shawn  
Powers  
runs  
through  
the latest  
issue.



system—er, I mean his phases of the moon script this month. When you start writing scripts with Dave, you start to realize all the cool additions you can make. Programming is like that. Dave shows how to figure out whether the moon is waxing or waning, which is really useful information when you're planning a family camping trip in werewolf territory.

Kyle Rankin continues his series on MCollective, which is server orchestration for configuration management systems. If Puppet and Chef can't quite handle the day-to-day needs of your environment, using MCollective might fill that need.

I explore Synology this month, which continues to fill my own needs for my home office. I've had multiple NAS machines through the years, and nothing has been as useful, reliable and performant as Synology. If you're looking for a storage device that also handles a plethora of server functions, you'll want to check out my column this issue.

der.hans has an incredible in-depth article this month on using password managers. We rely so much on our online accounts, it's vital that our passwords be strong and unique on each site. A password manager is quickly becoming the only feasible way to accomplish that. der.hans discusses how to manage passwords and retrieve them when needed. If password managers aren't convenient, it's unlikely anyone will use them, so learning the nuances of such an important technology is worth the effort.

Jan Newmarch finishes his series on low power wireless this month. The use of a low-powered wireless network is a constant reminder of how much our world is becoming connected. We're not far from a time when a hacker could infiltrate our toaster to ruin breakfast. Thankfully, Jan's series helps us to create our low powered wireless networks intelligently. Everyone should read his series, even if you never plan to implement something like 6LoWPAN.

We love bringing you the latest information from the Linux world, whether it's new products, fun apps or even in-depth security articles. Linux always has been at the forefront of computer security, and it's only going to stay there if we take security seriously and don't assume we're safe just because we use open source. If you're hoping to become a better, smarter Linux user, this issue is for you. ■



PREVIOUS  
Current\_Issue.tar.gz

NEXT  
UpFront



---

## Mars Lander Program

Regarding Dave Taylor's series on the Mars Lander game in the September, October and November 2016 issues: I appreciate the way he builds the program over three months in the magazine. Is there somewhere I can download the finished program, so I can peruse and play with it?

—John

*The finished program is available at <http://www.linuxjournal.com/files/linuxjournal.com/mars2.txt>.—Ed.*

## Simple Server Hardening

I found Kyle Rankin's "Simple Server Hardening" article in the October 2016 issue to be very useful, and I also look forward to more such tips, such as hardening of services that need to be on—for example, a mail server.

I have one comment: if we disable passwords altogether and allow access using SSH keys only, is there still a need to disable root login over SSH? The problem with reaching root via a non-root login (using sudo or SSH) is then your root is only as secure as such a non-root login. I personally don't even have a password for root, although I permit keys-only SSH logins to root. I would be glad if you could highlight risks with this arrangement, as against sudo and so on.

—Mayuresh

**Kyle Rankin replies:** *This is a great question. Yes, even if you disable password login altogether, you still should disable root login both over*

*SSH and locally, and use sudo instead. The core idea with requiring sudo for root access instead of using su is that it makes it easy to revoke any individual admin's access to root without having to change some central, shared password. Even if you require SSH keys for login, requiring sudo still provides extra protection in the case of a compromised key.*

*Attackers who have compromised an admin's SSH keys still have one extra barrier to the root account: they may be able to ssh in to a server as the admin using the admin's key (if the admin didn't take the extra step of password-protecting the keys), but the attackers still have to figure out the admin's personal password to be able to sudo to root. In your scenario, if your personal key were compromised, attackers would have direct access to root on all of your machines. Scenarios like this are why it's so important never to share personal keys between admin, password-protect your keys, pick good passwords for sudo, and never allow users to sudo up to root without their password.*

## Hodge Podge Comment

Shawn Powers' "Hodge Podge" article in the October 2016 issue was an interesting article, as usual.

In the article, he mentioned several economical cloud options. I wonder whether he has evaluated SDF.org (<https://sdf.org>). There are various tiers available to suit various needs, the most basic costing only a one-time fee for life on a shared cluster. There are VPS options as well at the higher end. How do these options figure in comparison with others that he mentioned?

—Mayuresh

# LINUX JOURNAL

## At Your Service

**SUBSCRIPTIONS:** *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an online digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your email address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly online: <http://www.linuxjournal.com/subs>. Email us at [subs@linuxjournal.com](mailto: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 online: <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](mailto:ads@linuxjournal.com) or +1 713-344-1956 ext. 2.

**Shawn Powers responds:** *I'd never heard of SDF.org, but when I visited the site, I couldn't find VPS options. I did see free shell access, which is cool, but that's all I could find.*

### Hodge Podge Suggestion

In Shawn Powers' "Hodge Podge" article in the October 2016 issue, he mentioned his love for Synology and GPS trackers. I think I share the same passion. In his quest for fun gadgets and programs, I would like to introduce Traccar. I use the free program called Traccar (<http://www.traccar.org>) to monitor the movements of my car. I installed the Traccar server on my Synology and the Traccar client on an old unused Android smartphone (with GPS). I installed the smartphone in my car and connected the smartphone to my car battery. On the smartphone, I installed a firewall app to minimize data traffic to and from Google (I'm not interested in updates or other stuff).

When the car is parked in the vicinity of my house, the smartphone uses the Wi-Fi of my AP. In this way, I minimize data traffic on my pre-paid data bundle. Only when I'm driving around GPS coordinates are sent to the Traccar server using the pre-paid data bundle. The client is set to an interval of one minute. Depending on the cost of your pre-paid data bundle and the amount of travel, you will use up \$10 in several months.

The Traccar server stores all coordinates (of multiple devices if needed) in a MySQL database. Via a web interface, you can locate your GPS device or see where the GPS device, in my case my car, travels or has traveled to.

Next, I wrote a C-program that does a MySQL query on the Traccar database. Whenever my car starts moving and moves beyond a pre-defined offset, the C-program sends an email, notifying me that the car is moving. The C-program also emails me when the car is stationary with its present position (street name and city). The C-program is running on a Raspberry Pi next to another beautiful program: Domoticz (<https://domoticz.com>).

There you go—a second life for your smartphone and lots of fun.

—Roland Horsten

**Shawn Powers responds:** *Very cool! I wonder if the program would record and upload only while within range of home Wi-Fi. It would be a great use for old phones and wouldn't require any cell service at all. Either way, I'll check it out. Thanks!*

### **Feedback on “Low Power Wireless: 6LoWPAN, IEEE802.15.4 and the Raspberry Pi”**

I'm one of the linux-wpan maintainers in the Linux kernel as well as the wpan-tools maintainer.

That this project was picked up for an article series in *Linux Journal* was a really nice surprise [see Jan Newmarch's articles in the November and December 2016 issues, as well as the final article in this issue].

Reading the first article, I have a few items I wanted to bring up:

- In the article Jan writes that the at86rf230 module needs to be loaded manually. That should not be needed and has not been in my testing so far. The device tree overlay he enabled should make sure that the driver is auto-loaded as well. Maybe that's something worth checking in his system.
- In the Python example, he uses TCP. I really would recommend using UDP for examples here. In his test setup, it will not make much of a difference, but in these networks, it can be really lossy and the TCP handshake makes things more complicated. In the end, you normally do not need a stream protocol for these types of networks.
- Using UDP for this example has the additional benefit that the kernel can apply next header compression (NHC, RFC6282) for UDP. If you use the right port range, that can save 2 bytes from the UDP header in each packet. (The port range is reduced to only 16 ports, starting from port 61617.)

A good article overall. Thanks for doing it and looking forward to the next episodes.

—Stefan Schmidt

## LETTERS

**Jan Newmarch responds:** *Thanks Stefan! Yes, you are right on both counts. I manually loaded the drivers in order to test if it was working. Once confirmed, reboots and the device tree load the drivers as needed. I should have mentioned that I loaded them manually only as a test and after that it no longer would be needed.*

*I was undecided as to whether to use TCP or UDP. UDP is the protocol of choice for low-power systems, but I thought that readers might be more familiar with TCP programs. The RPi can handle either; it has enough capabilities. In my third article (in this issue), I'm using the CoAP library aiocoap, and that is UDP-based.*

### Holy Triage, Batman!

I loved Susan Sons' "Holy Triage, Batman!" article in the November 2016 issue. So much good advice! Thank you!

—Mike

### Erratum

I just finished reading Reuven M. Lerner's At The Forge ("Preparing Data for Machine Learning") in the November 2016 issue, which was great and had me craving a burrito! But I noticed a typo: in the Resources section, the Data Science Weekly newsletter URL should be a .org and not a .com. The correct URL is <http://datascienceweekly.org>.

—Terrill

### SEND US YOUR PHOTOS

Send your Linux-related photos to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com), and we'll run the best ones here.

### WRITE LJ A LETTER

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

[RETURN TO CONTENTS](#)

# SUPERMICRO<sup>®</sup> MARKETPLACE

Powered by Silicon Mechanics



Broad  
Selection



Zero  
Defects



3-Year  
Warranty

**Your Source for  
Supermicro Platform Technology**

[Configure Now](#)



Talk to a Supermicro Expert!

866.352.1173



PREVIOUS  
Letters

NEXT  
Editors' Choice



## diff -u

### What's New in Kernel Development

**Anshuman Khandual** felt that there were a lot of devices with their own **RAM** in the world, and Linux should be able to allocate that memory to users as part of the general pool, rather than letting those devices hog it all for themselves.

It turns out to be hard to do this. Devices have to be initialized before they can be used, so the general pool of RAM would have to be grown by the kernel in a coherent way as new devices came online. Also, unlike regular system RAM, devices can be removed from the system without warning, which poses its own set of problems.

There always are solutions to these sorts of problems, but they may involve unpleasant compromises, such as increased code complexity, speed reduction or using some areas of RAM in certain ways but not others.

Sometimes there also are unexpected opportunities for collaboration that emerge from these sorts of proposals. In this case, Anshuman found that **Jerome Glisse's** work on **HMM** (heterogeneous memory management) might be relevant, because it had the similar issue of RAM not always being available to the general pool. Jerome suggested they work together on finding a proper solution.

**Dave Hansen** also pointed out that the existing projects **autonuma** (automatic non-uniform memory architecture) and **hugetlbfs** were



relevant to Anshuman's work, because they dealt with gathering up available RAM and adding it to the general memory pool as well.

But, not all projects can be merged in this way. Sometimes developers will make certain assumptions, or cut certain corners, because they don't see any other way to implement the thing they need. And, sometimes a developer on a similar project will need exactly the thing that the first developer had been trying to avoid. That's what happened here. **David Nellans**, who'd been working with Anshuman on his code, explained that, among other things, they wanted to avoid hugetlbfs support and concentrate on getting what they needed from **THP** (transparent huge tables) instead. But, Dave objected that this would force additional complexity into hugetlbfs—to special-case the RAM that Dave and Anshuman were working on.

There was quite a technical discussion, with no explicit resolution. This is the type of project that typically undergoes a wide-ranging discussion that gathers participants from all parts of the kernel. Eventually, various big-time kernel hackers may step in to adjust folks' direction and point out uncompromisable issues or extremely simple alternatives to complex debates.

A **ret2dir attack** is when a hostile user writes malicious code into kernel memory, hoping to trick the kernel into executing that code at a future date, like when a function call returns to that location. It's similar to the **ret2usr attack**, in which the same general thing occurs in user memory instead of kernel memory.

**Juerg Haefliger** recently wanted to implement

## THEY SAID IT

**Wait until it is night before saying that it has been a fine day.**

—**French Proverb**

**There's no such thing as quitting. Just sometimes there's a longer pause between relapses.**

—**Alan Moore**

**Self-confidence is the first requisite to great undertakings.**

—**Samuel Johnson**

**A moment's insight is sometimes worth a lifetime's experience.**

—**Oliver Wendall Holmes Jr.**

**Our lives begin to end the day we become silent about things that matter.**

—**Martin Luther King Jr.**

**eXclusive Page Frame Ownership** (XPFO), which would prevent a user from modifying RAM that hadn't been allocated explicitly to that user. He posted some code to do this and had a bit of a technical discussion about it with **Laura Abbott**. There were no serious objections—just suggestions regarding portability and locking efficiency. It looks like Juerg's code will go into the kernel at some point.

**Pavel Machek** felt that certain Linux systems—particularly smartphones—didn't do a great job when battery power became low. The **Nokia N900**, he said, would run down the battery until it didn't have enough power left to do a clean shutdown. It would just crash. Pavel proposed adding code to detect when the battery was too low and doing a proper shutdown in that case.

He posted some code to do this, but initially walked it back when **Tony Lindgren** pointed out that a more configurable patch would really be better. Specifically, he said the exact voltage that would trigger a shutdown should depend on what would be best for any given system. To do this, Pavel replied that a good approach would be to create a battery object in the device tree and create configurable properties for those objects.

On second thought, however, Pavel said that maybe configurability wasn't really the thing in this case. The hardware had its own mechanism for alerting the operating system that the battery was low or dead, and Pavel felt that Linux simply should respect those messages. There was no need to second guess the hardware in this case.

The discussion ended inconclusively, but presumably, some kind of clean shutdown will replace the current crash behavior.—Zack Brown

# The Best SharePoint 2016 and Office 365 Training!



**April 2-5, 2017 • AUSTIN, TEXAS**

SPTechCon offers classes and tutorials for IT professionals, business decision makers, information workers, developers and software and information architects. Each presenter at SPTechCon is a true SharePoint expert, with many drawn from Microsoft's tech teams or holding Microsoft MVP status.

Whether you're looking to upgrade to a more current version, making a move to the cloud, or simply need answers to those daunting problems you've been unable to overcome, SPTechCon is the place for you! Come join us!

- Choose from more than 80 classes and panel sessions
- Improve your skills and broaden your knowledge of Microsoft's collaboration and productivity software
- Learn about SharePoint 2016, the latest on-premises server release from Microsoft
- Tips and tricks for working with SharePoint 2013 and 2010, and Office 365
- Practical information you can put to use on the job right away!
- The most knowledgeable instructors working in SharePoint today

[www.sptechcon.com](http://www.sptechcon.com)

# Android Candy: the Verbification of Video Chat

People who study the history of languages probably will look back at our current time



Google Duo

and scratch their heads. We keep inventing verbs! First, Google became the verb we use for searching. Then, “Facebooking” someone became a viable way to contact them. Heck, I forgot about “texting” someone. It seems we just keep taking perfectly good nouns and making them verbs. We keep verbing all our nouns! But I digress.

Unfortunately, I’ve noticed a trend where people who want to do video chat are starting to say, “Facetime me.” The problem isn’t with the terminology, it’s with the proprietary technology. If you’re an Android user, you can’t “Facetime” someone. Google Hangouts sort of solves that problem, but it’s not as user-friendly as Facetime on iOS is.

Google hopes to change that with its new Duo app. It’s designed very much like Facetime, but it has native iOS and Android apps. In my trials, the video quality is fine, so it’s really only adoption that needs to happen in order to become the standard across platforms. It’s worth a try, but you’ll have to convince your Apple friends to install a new app for video chatting. Also, “duo” doesn’t really lend itself to verbification, so I’m a little worried Google has a non-starter on that note alone!

—Shawn Powers



Where every interaction matters.

# break down your innovation barriers

**power your business to its full potential**

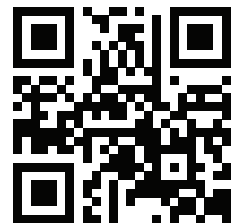
When you're presented with new opportunities, you want to focus on turning them into successes, not whether your IT solution can support them.

Peer 1 Hosting powers your business with our wholly owned FastFiber Network™, global footprint, and offers professionally managed public and private cloud solutions that are secure, scalable, and customized for your business.

Unsurpassed performance and reliability help build your business foundation to be rock-solid, ready for high growth, and deliver the fast user experience your customers expect.

**Want more on cloud?**

**Call: 844.855.6655 | [go.peer1.com/linux](https://go.peer1.com/linux) | [View Cloud Webinar:](#)**



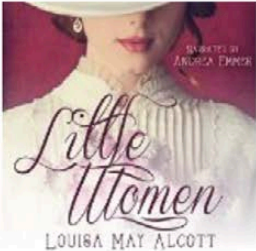
---

**Public and Private Cloud | Managed Hosting | Dedicated Hosting | Colocation**

# Listen To Me Cheaply

I listen to a lot of books. A lot. And honestly, although I've written about the "Listen" app for audiobooks, I tend to use Audible more than anything else anymore. Part of the reason is the Android app finally has more fine-grained speed settings. (I prefer around 1.4x speed.) iPhone people don't have that seemingly simple feature. Just saying.

The main reason I've been using Audible, however, is the pricing. Yes, I'm a Platinum subscriber, so I get two credits a month. But, it's the "add narration" feature on the Kindle store that really does it



**Best Seller**

AUDIOBOOK

**Little Women**

UNABRIDGED

By [Louisa May Alcott](#)

Narrated By [Andrea Emmes](#)

*Whispersync for Voice-ready*

Length: 17 hrs and 36 mins

Release Date: 08-16-16

★★★★★ 4.6 (27 ratings)


Regular Price: ~~\$31.42~~

Member Price: **\$21.99** or 1 Credit

[Add to Cart](#)

[Add to Wish List](#)

[Give as a Gift](#)

 SAMPLE

Back to search results for "little women"



**Little Women**

LOUISA MAY ALCOTT

A PUBLIC DOMAIN BOOK


 Audible Narration

**Little Women Kindle Edition**

by [Louisa May Alcott](#) (Author)

★★★★★ 1,184 customer reviews

See all 6 formats and editions Amazon Exclusive

<b>Kindle</b> \$0.00	<b>Hardcover</b> \$22.99 
-------------------------	---

Read with Our **Free App**     5 Used from \$19.00  
10 New from \$22.99

Following the lives of four sisters on a journey out of adolescence, Louisa May Alcott's *Little Women* explores the difficulties associated with gender roles in a Post-Civil War America.

Print List Price: ~~\$40.53~~

Kindle Price: **\$0.00**

Save \$40.53 (100%)

 Buy now with 1-Click®


Add Audible narration to your purchase for just **\$2.99**

Deliver to:

Donna's Kindle ▾

Add to List ▾

[Enter a promotion code or Gift Card](#)

[Share](#)     [<Embed>](#)

though. It's often cheaper to buy the Kindle version of a book and add Audible narration for less than the cost of the audiobook alone! That's not always the case, but it's always worth checking. Look at my example in the screenshots.

Yes, this book is free. But, you can add Audible Narration for \$2.99. Notice how much it would cost to buy the book outright: \$21.99! Not all books come with such a good deal, but many do. And if you add Audible narration, the book is added to your Audible account like any other book. You never have to read the Kindle version, but it's still nice to have both—especially when it's cheaper than just buying the audiobook!—Shawn Powers

**LINUX**<sup>™</sup>  
JOURNAL



## LINUX JOURNAL ARCHIVES

Issues 1–272

The First 23 Years of Linux Journal (1994–2016)

Archive  
1994–2016  
**NOW  
AVAILABLE!**

**SAVE \$10.00**

by using  
discount code  
**2017ARCH**  
at checkout.

Coupon code  
expires 3/28/2017

[www.linuxjournal.com/archive](http://www.linuxjournal.com/archive)

# Never Trust Yellow Fruit

You've probably heard about the WiFi Pineapple from Hak5. It's a fascinating device that allows you to do some creepy pen testing. It's the sort of tool that could be used for evil, but it's also incredibly useful for securing networks.



*WiFi Pineapple*

The hardware is fairly basic and resembles an off-the-shelf router. The multiple network interfaces really shine, however, when paired with the operating system. The WiFi Pineapple software creates a rogue, hidden access point that purposefully tricks clients into connecting to *it* instead of the AP they're usually connected to.





It doesn't stop there, either. The software sniffs the wireless network for clients looking for APs they know, and then creates fake SSIDs that match what clients are looking for. Once the client associates with the SSID it thinks it already knows, the Pineapple provides internet service, but it also sniffs every packet along the way. Scary stuff.

There are many uses for the WiFi Pineapple, and many of them are white-hat sorts of things. Be warned though, because you might be connected to someone else's WiFi Pineapple right now! To get your own piece of dangerous awesomeness, head over to <http://wifipineapple.com> now. There is a 2.4GHz-only version as well as a dual-frequency model.

—Shawn Powers

## LINUX JOURNAL

on your  
e-Reader

Customized  
Kindle and Nook  
editions  
available

LEARN MORE



# Analyzing Videos for Fun and Profit

People's phones and all of the various sensors that may be built in to them is a source of scientific data logging that almost everyone carries around. Although the selection of sensors varies from phone to phone, they almost all have a camera. In this article, I take a look at a piece of software called Tracker that can be used to analyze videos you take of experiments.

You can download Tracker directly from the project page: <http://physlets.org/tracker>. A lot of good documentation is available at the main website, including examples of how you might be able to use it from the community of other users. Tracker is written in Java, however, so you also need to have a JVM installed on your system before you can use it. For example, on a Debian-based system, you can install a very good JVM with the command:

```
sudo apt-get install openjdk-8-jre
```

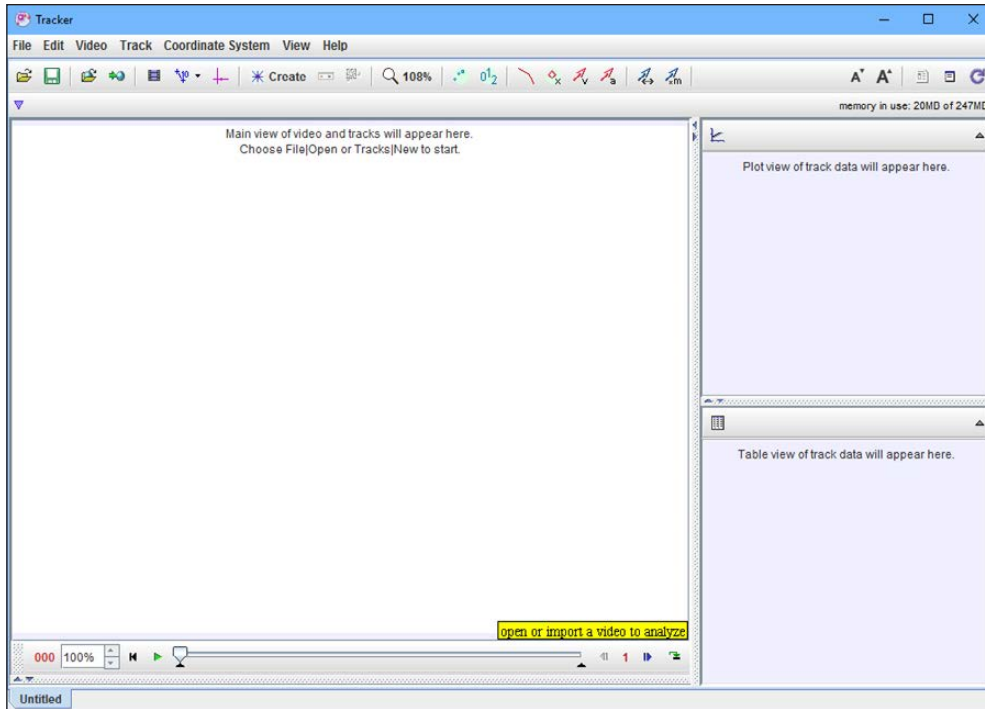
You then can download the relevant installer for your system and run it from a terminal window. You probably will need to make the installer executable with a command like this:

```
chmod +x Tracker-4.95-linux-64bit-installer.run
```

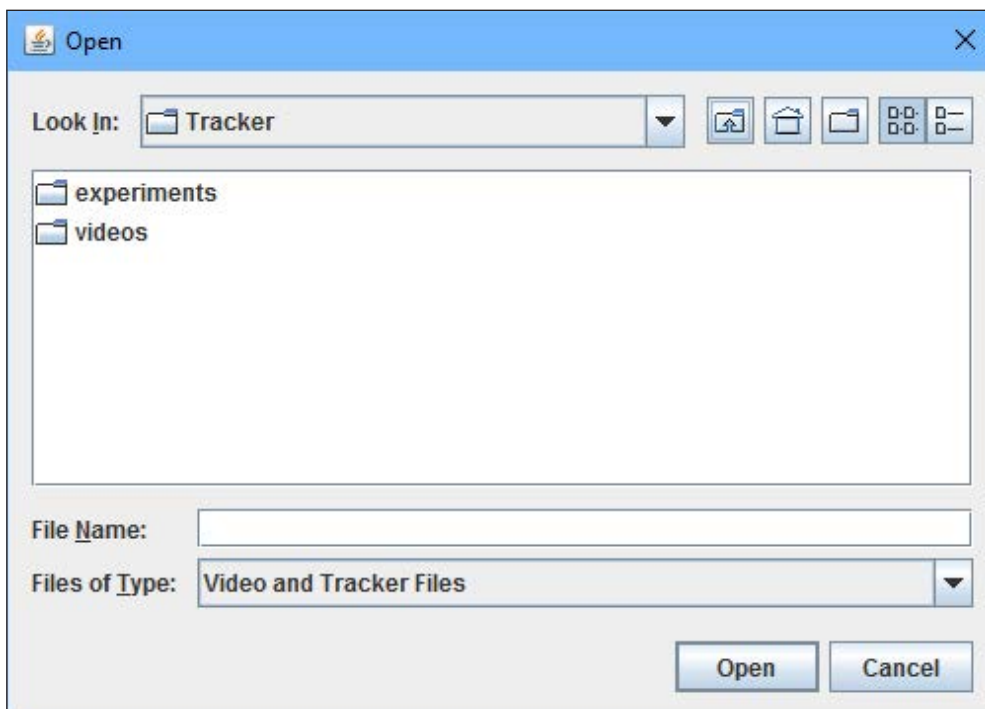
Don't forget to use the sudo command so that you have the correct permissions to do the installation.

Once it's installed, you should have a new entry in your application's menu system. Starting Tracker will give you a new empty project for beginning your video analysis.

Although you could start analyzing your own videos immediately, you may want to use simpler videos while learning how Tracker works. When you ran the initial installer for Tracker, it should have asked you whether you also wanted to install example files. Assuming you said yes, you now



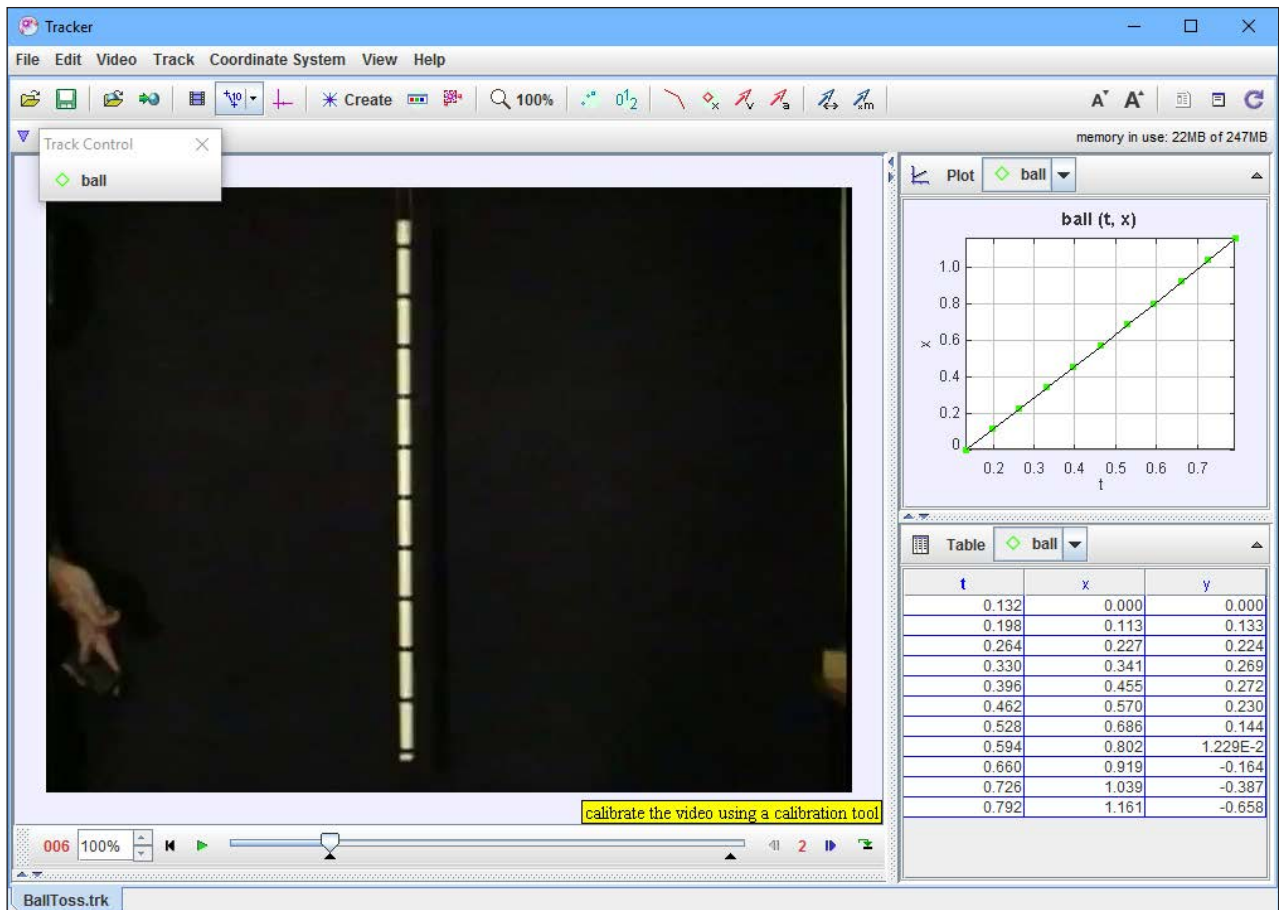
**Figure 1.**  
When you start Tracker, you get a new empty project.



**Figure 2.**  
The open file dialog box allows you to open movie files or Tracker files.

can load one of those within Tracker. When you click on the menu item File→Open File, you'll see a new window where you can select either a movie file or a Tracker file.

To begin with, let's look at one of the Tracker files from the experiments folder. In Figure 3, I have loaded the experiment file named BallToss.trk.



**Figure 3. Loading a Tracker file opens all of the parts of a previous analysis.**

The main window displays the movie that is being analyzed as part of the experiment. The pane in the bottom right-hand side shows a table of  $x$  and  $y$  coordinates for a series of time units. A similar table is generated for each tracked object within the movie. In this example experiment, the only tracked object is the ball. The top right-hand pane contains the associated plot for the data stored in the table. This way, you can have a visual representation of the tracked motion.

So, what can Tracker actually track? At the bottom of the movie display is a set of controls. If you click on the green arrow, the movie will start to play. As it does, Tracker actually registers the movement of the tracked object and also updates the data panes on the right-hand side.

Although these panes do give you lots of raw data, there really isn't any analysis being done yet. Clicking the View→Data Tool menu item pops open a new window where you can use this raw data to figure out what

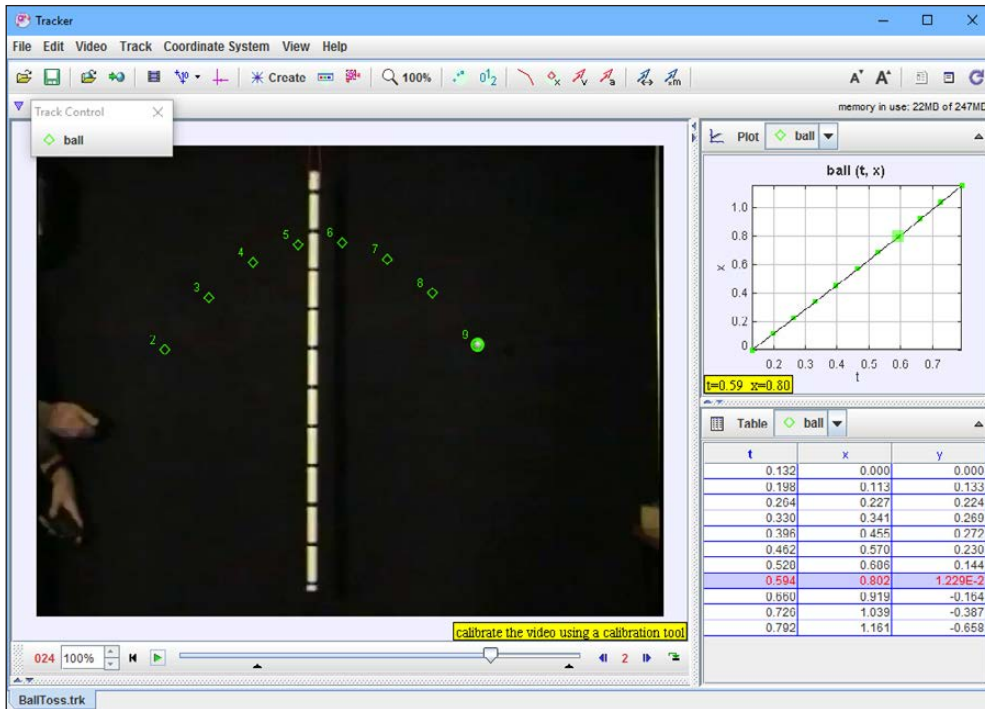


Figure 4. As the movie progresses, Tracker updates the data panes on the right-hand side for the tracked object.

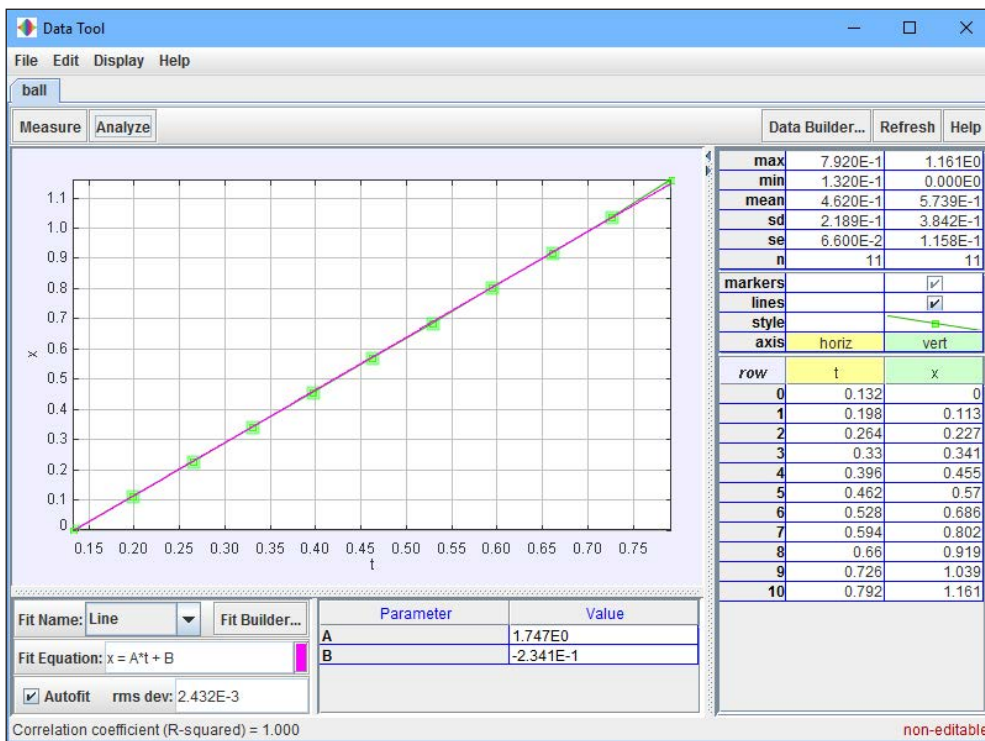
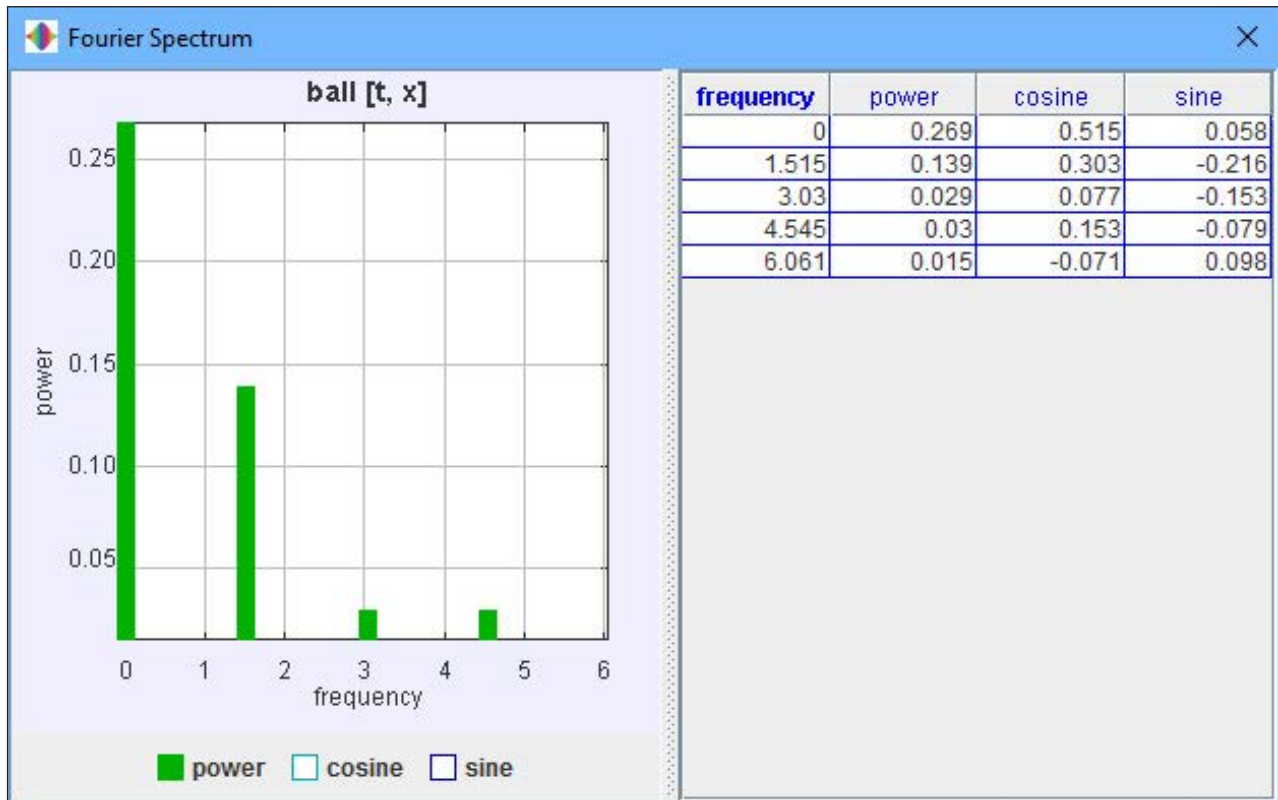


Figure 5. The Data Tool window provides a set of analysis tools to look at the raw data pulled from the video.

is happening in the video.

The Measure and Analyze buttons at the top of the plot window provide a set of different sections that you can display. This gives you several different analysis options for your data. Clicking the Measure button



**Figure 6. Tracker also can do a Fourier analysis of the observed motion.**

gives you the options Coordinates, Slope and Area. When you select one or more of those options, values are displayed at the bottom of the plot representing each of the selected options.

When you click the Analyze button, you get a drop-down with the Statistics, Curve Fits and Fourier Spectrum options. Each of those options actually opens a new pane in the Data Tool window. The exception is the Fourier Spectrum option. This option actually opens a new window where a Fourier analysis is done on the raw data from the tracked object.

The Statistics option gives the data displayed in the top right pane. This is the set of descriptive statistics—that is, the maximum, minimum, mean and standard deviation, among other values.

Selecting the Curve Fits option gives you the pane at the bottom of the plot window. Here you can select what type of curve is being fitted to your raw data. You can use a number of default functions to try to fit your data. In this case, the linear function is the obvious choice for a fitting function. You can see the function being used,  $x=A*t+B$  in

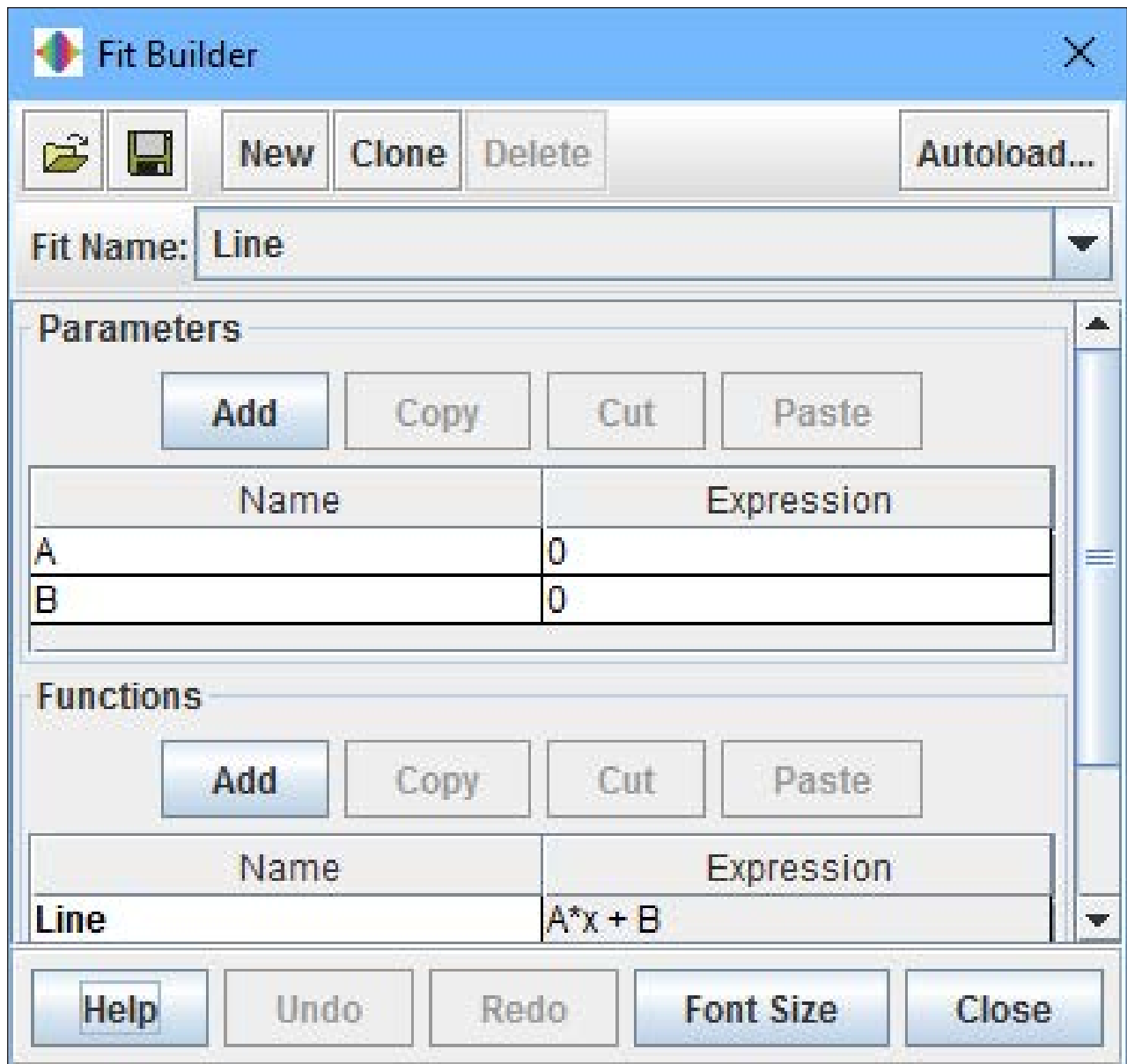


Figure 7. You define your own fitting functions as part of the data analysis.

this case, along with the calculated values for the parameters A and B. If you are tracking an object that has a really odd motion, you can click the Fit Builder button to pop up a new window where you define a new fitting function.

What do you do as part of a new project? Let's say you want to analyze the pendulum video from the sample video folder. You can set some calibration items within the video by either selecting the relevant buttons

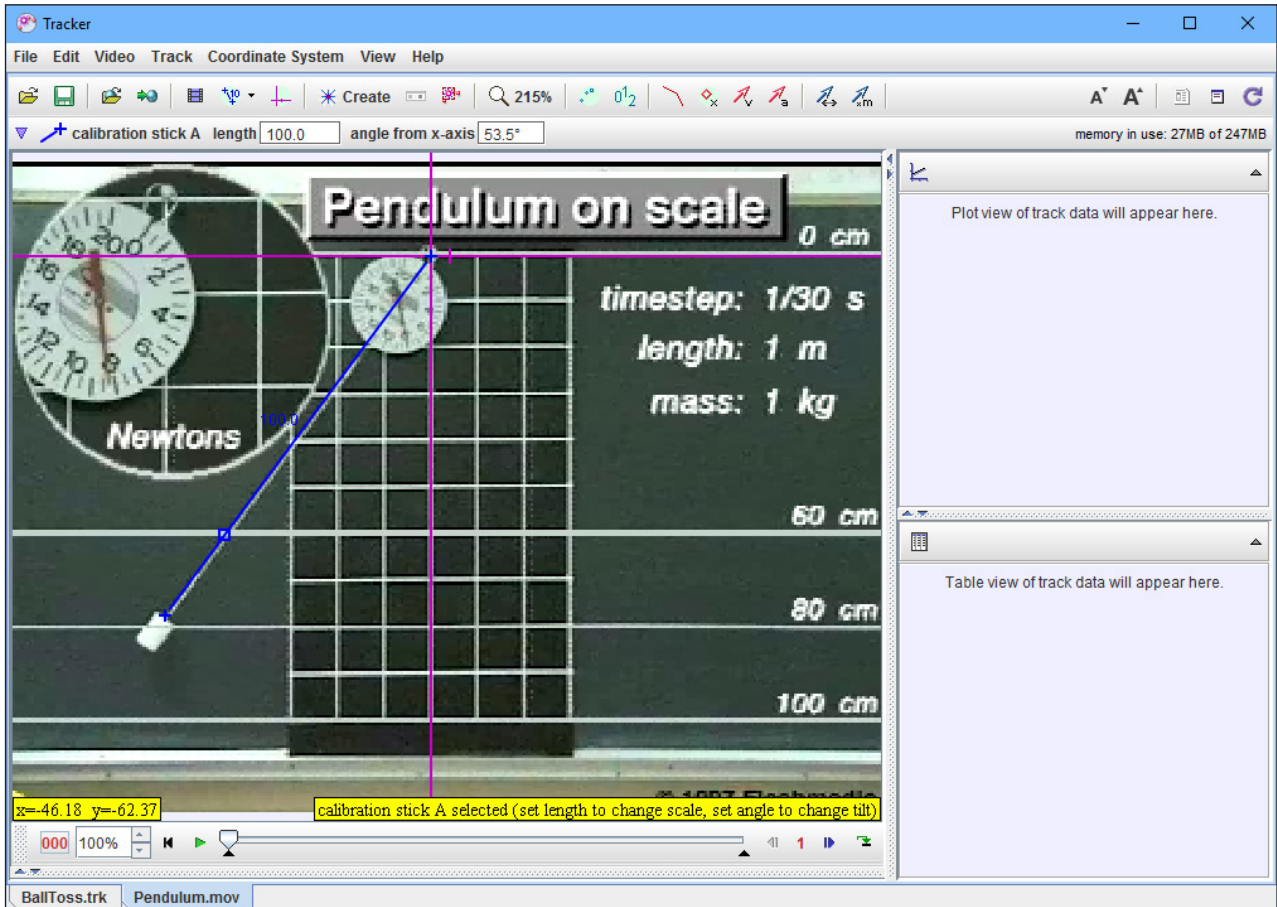


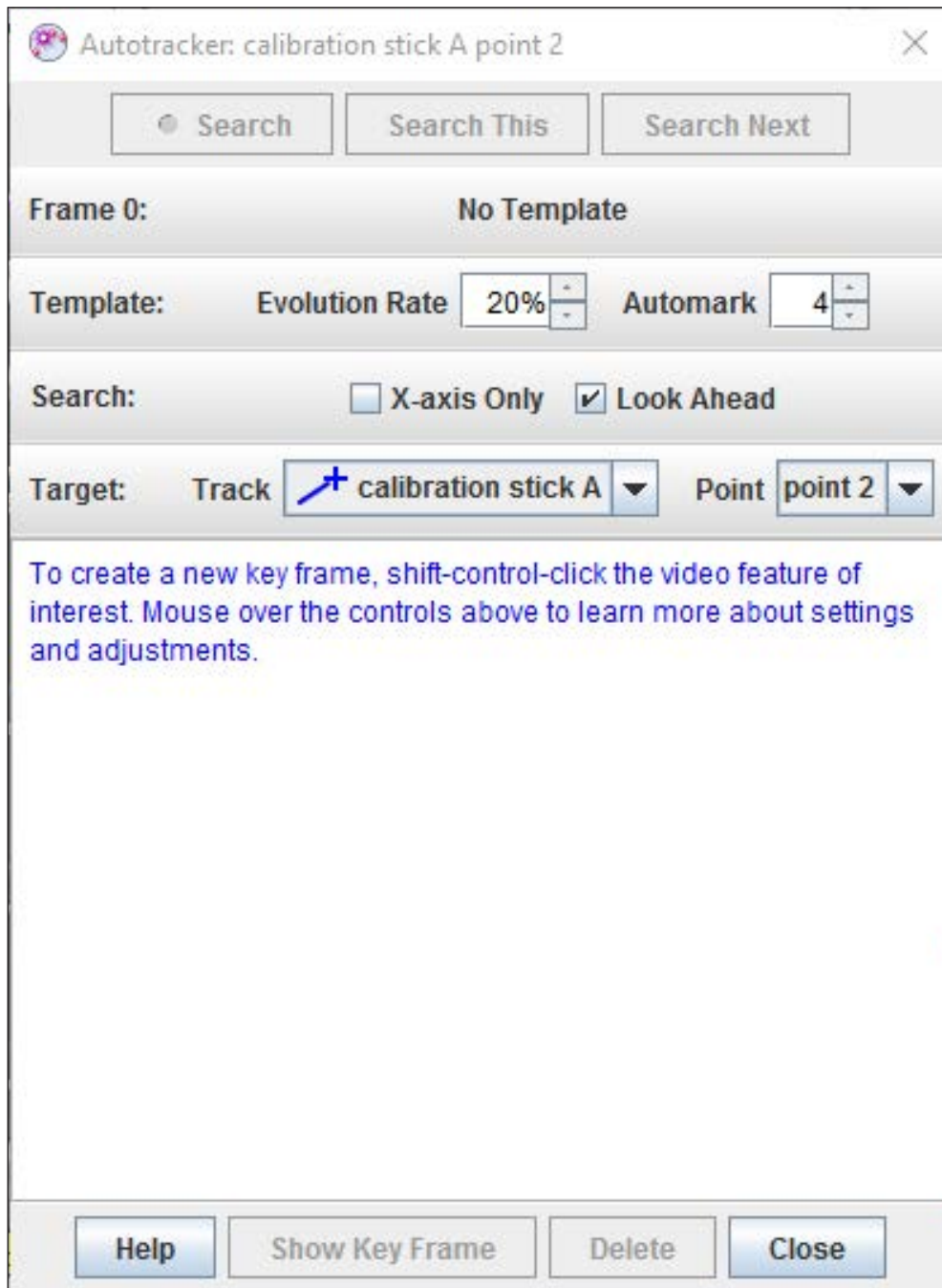
Figure 8. You can set up calibration tools within your video to be analyzed.

at the top of the window or selecting the items under the menu entry Track→New→Calibration Tools. You can set up items like a calibration stick or a reference set of axes.

One way to do the analysis is to step through the video, frame by frame, selecting the object to be tracked within each frame. This may be the only way to collect the raw data, but humans are lazy. If the object being tracked is relatively consistent across the duration of the video clip in question, you can try using the autotracker. Clicking the autotracker button will pop up a new window where you can control what is being tracked.

Tracker helpfully displays reminder information on how to use the software. In this case, it reminds you that you need to press Ctrl, Shift and left-click the mouse button on the object of interest. If you then





**Figure 9.**  
The autotracker tries to follow an object from frame to frame automatically.

click the search button within the autotracker window, it will move through each frame, looking for the object in question. Once it is done, you should have a full set of data from the tracked object. From the plot in Figure 10, you can see right away that the pendulum follows the expected motion. You also can track multiple objects by creating a new track for each object.

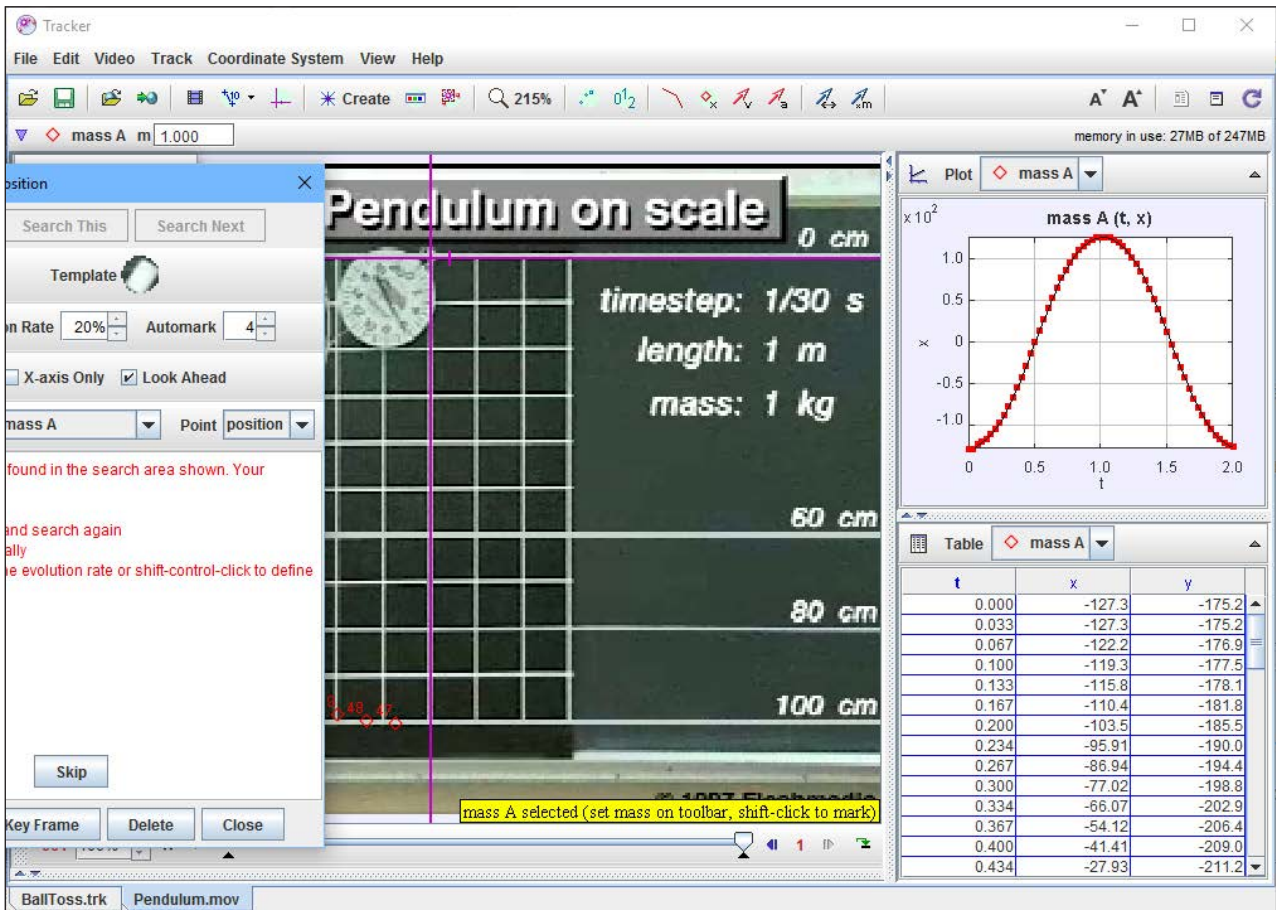


Figure 10. The autotracker can generate all of your raw data automatically.

Now that you've taken a quick look at Tracker and the kind of analysis you can do with it, you should be comfortable enough to start experimenting with it. Don't forget to share any interesting ideas you come up with on the community page and add to a great resource for citizen science.

—Joey Bernard

RETURN TO CONTENTS

The Fifteenth Annual  
Southern California Linux Expo

# SCALE 15x

March 2-5, 2017

Pasadena Convention Center  
Pasadena, CA



<http://www.socallinuxexpo.org>  
Use Promo Code LJ15X for a 30%  
discount on admission to SCALE

◀ PREVIOUS  
UpFront

NEXT ▶  
Reuven M. Lerner's  
At the Forge

## Low Tech High Tech



Google Cardboard should be terrible. Really, it should. It's literally made of cardboard. I remember as a kid some cereal boxes came with spy glasses you had to cut out of the box itself—and they were terrible. But Google Cardboard is amazing. Granted, you need to add your \$750 Android



phone to it, but that's already in your pocket anyway.

The reason Google Cardboard can be so simple is that its only job is to let you focus on a phone that is two inches from your face. A couple plastic lenses held in place by a chunk of cardboard isn't terribly high tech. But once you have that technology in place, Google Cardboard lets you literally transport to another world. Using apps from the Google Play Store, the VR experience is shockingly good using nothing more than a modern Android cell phone.

The truth is, it takes a lot of graphical power to render two separate dynamic videos on a screen at one time. (That's how it makes the experience 3D.) So whatever VR app you download will likely have a slightly blocky-looking world. The crazy part is, that doesn't detract from the experience. In fact, some blocky trees almost make the experience that much more surreal because it's *obvious* you're in a simulation. Yet, when you turn your head, it changes like you're actually there!

If you want to experience some pretty cool virtual reality, but don't want to invest in one of the expensive, Windows-centric systems like the HTC Vive, give Google Cardboard a try. You can get the actual cardboard goggles on Amazon, or you could spend a few more dollars to get a plastic set that is a little more sturdy. Either way, the experience is amazing! It's so amazing, in fact, that Google Cardboard gets the Editors' Choice award this month. Virtual reality using the same technology that transports pizza? Yes please!

—Shawn Powers

[RETURN TO CONTENTS](#)

# Testing Models

You've created a machine-learning model. Now, how do you know if it works well?



**REUVEN M. LERNER**

Reuven M. Lerner offers training in Python, Git and PostgreSQL to companies around the world. He blogs at <http://blog.lerner.co.il>, tweets at @reuvenmlerner and curates <http://DailyTechVideo.com>. Reuven lives in Modi'in, Israel, with his wife and three children.

◀ PREVIOUS  
Editors' Choice

NEXT  
Dave Taylor's  
Work the Shell ▶

**IN MY LAST FEW ARTICLES**, I've been dipping into the waters of "machine learning"—a powerful idea that has been moving steadily into the mainstream of computing, and that has the potential to change lives in numerous ways. The goal of machine learning is to produce a "model"—a piece of software that can make predictions with new data based on what it has learned from old data.

One common type of problem that machine learning can help solve is classification. Given some new data, how can you categorize it? For example, if you're a credit-card company, and you have data about a new purchase, does the purchase appear to be legitimate or fraudulent? The degree to which you can categorize a purchase accurately depends on the quality of your model. And, the quality of your model will generally depend on not only the algorithm you choose, but also

the quantity and quality of data you use to “train” that model.

Implied in the above statement is that given the same input data, different algorithms can produce different results. For this reason, it’s not enough to choose a machine-learning algorithm. You also must test the resulting model and compare its quality against other models as well.

So in this article, I explore the notion of testing models. I show how Python’s scikit-learn package, which you can use to build and train models, also provides the ability to test them. I also describe how scikit-learn provides tools to compare model effectiveness.

### Testing Models

What does it even mean to “test” a model? After all, if you have built a model based on available data, doesn’t it make sense that the model will work with future data?

Perhaps, but you need to check, just to be sure. Perhaps the algorithm isn’t quite appropriate for the type of data you’re examining, or perhaps there wasn’t enough data to train the model well. Or, perhaps the data was flawed and, thus, didn’t train the model effectively.

But, one of the biggest problems with modeling is that of “overfitting”. Overfitting means that the model does a great job of describing the training data, but that it is tied to the training data so closely and specifically, it cannot be generalized further.

For example, let’s assume that a credit-card company wants to model fraud. You know that in a large number of cases, people use credit cards to buy expensive electronics. An overfit model wouldn’t just give extra weight to someone buying expensive electronics in its determination of fraud; it might look at the exact price, location and type of electronics being bought. In other words, the model will precisely describe what has happened in the past, limiting its ability to generalize and predict the future.

Imagine if you could read letters that were only from a font you had previously learned, and you can further understand the limitations of overfitting.

How do you avoid overfit models? You check them with a variety of input data. If the model performs well with a number of different inputs, it should work well with a number of outputs.

In my last article, I continued to look at data from a semi-humorous study in which evaluations were made of burritos at a variety of restaurants in Southern California. Examining this data allowed one to identify which elements of a burrito were important (or not) in the overall burrito's quality assessment. Here, in summary, are the steps I took inside a Jupyter notebook window in order to create and assess the data:

```
%pylab inline
import pandas as pd                # load pandas with an alias
from pandas import Series, DataFrame # load useful Pandas classes
df = pd.read_csv('burrito.csv')    # read into a data frame

burrito_data = df[range(11,24)]
burrito_data.drop(['Circum', 'Volume', 'Length'], axis=1, inplace=True)
burrito_data.dropna(inplace=True, axis=0)

y = burrito_data['overall']
X = burrito_data.drop(['overall'], axis=1)

from sklearn.neighbors import KNeighborsRegressor # import
                                                # classifier
KNNR = KNeighborsRegressor()                   # create a model
KNNR.fit(X, y)                                 # train the model
```

So, is the model good or not? You can know only if you try to make some predictions for which you know the answers, and see whether the model predicts things correctly.

Where can you find data about which you already know the answers? In the input data, of course! You can ask the model (KNNR) to make predictions about X and compare those with y. If the model were performing categorization, you even could examine it by hand to get a basic assessment. But using regression, or even a large-scale categorization model, you're going to need a more serious set of metrics.

Fortunately, scikit-learn comes with a number of metrics you can use. If you say:

```
from sklearn import metrics
```



then you have access to methods that can be used to compare your predicted values (that is, from the original “y” vector) to the values that were computed by the model. You can apply several scores to the model; one of them would be the “explained variance score”. You can get that as follows:

```
y_test = KNR.predict(X)

from sklearn import metrics
metrics.mean_squared_error(y_test, y)
```

Notice what’s happening here. You’re reusing the input matrix X, asking the model to predict its outputs. But, you already know those outputs; those are in y. So now you see how closely the model comes to predicting outputs that already were fed into it.

On my system, I get 0.64538408898281541. Ideally, with a perfect model, you would get a 1, which means that the model is okay, but not amazing.

However, at least you now have a way of evaluating the model and comparing it against other models that might be better or worse. You even can run KNR for different numbers of neighbors and see how well (or poorly) each model does:

```
for k in range(1,10):
    print(k)
    KNR = KNeighborsRegressor(n_neighbors=k)
    KNR.fit(X, y)
    y_test = KNR.predict(X)
    print "\t", metrics.mean_squared_error(y_test, y)
    print "\t", metrics.explained_variance_score(y_test, y)
```

The good news is that you have now looked at how the KNR model changes when configured with different values of `n_neighbors`. Moreover, you see that when `n_neighbors = 1`, you get no error and 100% explained variance. The model is a success!

## Split Testing

But wait. The above test is a bit silly. If you test the model using data that was part of the training, you would be surprised if the model *didn't* get it at least partly right. The real test of a model is how well it works when it encounters new data.

It's a bit of a dilemma. You want to test the model with real-world data, but if you do that, you don't necessarily know what answer should appear. And, that means you can't really test it after all.

The modeling world has a simple solution to this problem. Use only a subset of the training data to train the model, and use the rest for testing it.

scikit-learn has functionality that supports this "train-test-split" functionality. You invoke the `train_test_split` function on your original X and y values, getting two X values (for training and testing) and two y values (for training and testing) back. As you might expect, you then can train the model with the `X_train` and `y_train` values and test it with `X_test` and `y_test`:

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
    ↳test_size=0.25)
```

```
KNR = KNeighborsRegressor(n_neighbors=1)
KNR.fit(X_train, y_train)
y_pred = KNR.predict(X_test)
print "\t", metrics.mean_squared_error(y_test, y_pred)
print "\t", metrics.explained_variance_score(y_test, y_pred)
```

Suddenly, this amazing model no longer seems so amazing. By checking it against values it hadn't seen before, it's giving a mean squared error of 0.4 and an explained variance of 0.2.

This doesn't mean the model is terrible, but it does mean you might want to check it a bit further. Perhaps you should (again) check additional values of `n_neighbors`. Or, perhaps you should try something other than `KNeighborsRegressor`. Again though, the key takeaway is that you are now using a real, reasonable way to evaluate that model, rather than just eyeballing the numbers and assuming (hoping) that all is well.

**What you really need to do is try different splits, so you can be sure that no matter what training data you use, the model performs optimally.**

## Multiple Splits

The split-test that you do might somehow tickle the model in such a way that it gives particularly good (or bad) results. What you really need to do is try different splits, so you can be sure that no matter what training data you use, the model performs optimally. Then, you can average the results over a bunch of different splits.

In the world of scikit-learn, this is done using `KFold`. You indicate how many different instances of the model you'll want to create and the number of "folds" (that is, split tests) you'll want to run:

```
from sklearn.cross_validation import KFold, cross_val_score
kfold = KFold(n=len(X), n_folds=10)
```

With the `kfold` object in place, you then can pass it to the `cross_val_score` method in the `cross_validation` module. You pass it the model (KNN, in this case), `X`, `y` and the `kfold` object you created:

```
v_results = cross_val_score(KNN, X, y, cv=kfold)
```

The `cv_results` object you get back describes the cross validation and typically is analyzed by looking at its mean (that is, what was the average score across those runs) and the standard deviation (that is, how much variance was there across runs):

```
print cv_results.mean()
print cv_results.std()
```

In this particular case, the results aren't that promising:

```
0.310254620082
```

```
0.278746712239
```

In other words, although the `n_neighbors=1` seemed to be so terrific when first analyzed, using all of the training data for testing, that no longer appears to be the case.

Even if you stick with KNR as your classifier, you still can incorporate KFold, checking to see when (if) a different value of `n_neighbors` might be better than the value of 1 you gave here:

```
from sklearn.cross_validation import KFold, cross_val_score
```

```
for k in range(1,10):
```

```
    print(k)
```

```
    KNR = KNeighborsRegressor(n_neighbors=k)
```

```
    kfold = KFold(n=len(X), n_folds=10)
```

```
    cv_results = cross_val_score(KNR, X, y, cv=kfold)
```

```
    print "\t", cv_results.mean()
```

```
    print "\t", cv_results.std()
```

Sure enough, when `k=9`, you get results that are significantly better than when `k=1`:

```
0.594573190846
```

```
0.161443573949
```

That said, I do believe it's likely you can create a better model. Perhaps a better classifier for regression would improve things. Perhaps using categorization, rather than regression, in which you round the values in `y` to the nearest integer and treat scores as 5 distinct categories, would work. Perhaps, as mentioned before, I should have paid more attention to which columns were most (and least) important and done some better feature selection.

Regardless, with a proper test system in place, you're now able

to start tackling these questions intelligently with a way to evaluate your progress.

### Summary

It's not enough to create a machine-learning model; testing it is also important. As you saw here, scikit-learn makes it relatively easy to create, split-test and then evaluate one model or even a whole bunch of them.

Supervised learning isn't the only type of machine learning out there. In many cases, you can ask the computer to divide your data into multiple groups based on heuristics it develops, rather than categories that you have trained. In my next article, I plan to look at how (and when) to build "unsupervised learning" models. ■

Send comments or feedback via  
<http://www.linuxjournal.com/contact>  
or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

[RETURN TO CONTENTS](#)

## Is the Moon Waxing or Waning?

Dave finalizes his phase of the moon script.



DAVE TAYLOR

Dave Taylor has been hacking shell scripts on UNIX and Linux systems for a really long time. He's the author of *Learning Unix for Mac OS X* and *Wicked Cool Shell Scripts*. He can be found on Twitter as @DaveTaylor, and you can reach him through his tech Q&A site: <http://www.AskDaveTaylor.com>.

---

PREVIOUS

◀ Reuven M. Lerner's  
At the Forge

NEXT

Kyle Rankin's  
Hack and / ▶

---

**I DON'T KNOW ABOUT YOU**, but I'm still thinking extra-planetary thoughts as we go through the tail end of this particularly contentious election season and its aftermath. Maybe life on other planets is easier? Ah, maybe not.

In any case, I completed the Martian lander and now am enmeshed in a phase of the moon script. In my last article, I talked about the complications of calculating the phase of the moon and decided simply to scrape the same web site that Google uses: <http://www.moongiant.com>.

That site provides the current moon illumination level, which lets you break it down into the phases of new moon, crescent, quarter, gibbous and

**Therefore, to ascertain waxing or waning, all you need to do is know the moon's illumination level today and either yesterday or tomorrow.**

full. Amateur astronomers know that the fun part of tracking the moon's phase is to understand whether it's "waxing" (growing more illuminated) or "waning" (growing less illuminated).

Although at any given moment the moon is illuminated based on its location, and your location, relative to the sun, the full cycle of a moon phase starts and ends with a new (0% illuminated) moon, and the full moon (100% illuminated) is the mid-point of the journey.

Therefore, to ascertain waxing or waning, all you need to do is know the moon's illumination level today and either yesterday or tomorrow. Fortunately, the Moon Giant website obligingly has the ability for you to ascertain the illumination level for a specific date.

A quick visit to the site with a regular web browser reveals that it works using a date-based URL format like this: <http://www.moongiant.com/phase/11/6/2016>.

So, you can build the date URL for the day before today with a call to the date program. If you've got the GNU version of date, it's easy to back up a day:

```
$ date
Mon Nov  7 11:40:31 MST 2016
$ date -v -1d
Sun Nov  6 11:40:15 MST 2016
```

It turns out that you also can specify that you want to back up 24 hours, although, of course, the net result is the same:

```
$ date -v -24H
Sun Nov  6 11:40:24 MST 2016
```

## WORK THE SHELL

More important, you can pass `date` a format string that you then can evaluate with the `eval` function, so you can set month, day and year for yesterday in one easy step:

```
$ eval $( date -v -1d +"mon=%m day=%d year=%Y" )  
$ echo month = $mon, day = $day and year $year  
month = 11, day = 06 and year 2016
```

It's quite a handy trick when you need to work with extracting specific elements from `date` and 10x that when it also involves date math.

### Older Date Programs Are More Complicated

But, what if your version of `date` doesn't include the `-v` flag and doesn't have all these fancy features? Then, my friend, you are facing a definite challenge. Date math is pretty easy, except for the edge cases.

That is, it's easy to extract the current month, day and year from even the most rudimentary Linux version of `date`, and it's obviously easy to subtract one from the day, but what if it's the first of the month? Or the first of the year?

That's doable too, but it's just a bit more work. Notably, you'll also want to know about leap years, because one day prior to March 1, 2016, might be February 28, or it might be February 29, depending on whether 2016 was a leap year.

Now a sneaky way to do it simply would be to sidestep the issue. If the day number of the month is greater than 1, subtract one to get yesterday's date. If it is the first, however, *add one and reverse the logic of the waxing/waning test*.

Fortunately, I do have the more sophisticated `date` program, so I'm going to do that most frustrating of things and leave this particular facet as the proverbial exercise for the reader.

### Yesterday's Lunar Illumination Level

Knowing the format of the Moon Giant URL when you specify a date, and knowing how to use `eval` and `date` to get the numeric month, day and year values for yesterday, here's some code to put



**But wait, “waxing” and “waning” applies only to crescent and gibbous moon phases. If the moon is new, quarter or full, neither word applies in common astronomical parlance.**

that all together:

```
url_ago="http://www.moongiant.com/phase"
eval $( date -v -1d +"mon=%m day=%d year=%Y" )
ydayurl="$url_ago/$mon/$day/$year"
```

The good news is that the resultant web page has the same format as the page for today’s illumination level too, so the same `curl|grep` sequence explored in my last article can be reused for this task:

```
yillumlevel="$( curl -s "$ydayurl" | grep "$pattern" | tr ', ' '\
' | grep "$pattern" | sed 's/[^0-9]//g')"
```

In fact, let’s add a debugging statement that displays both today’s lunar illumination level and yesterday’s level:

```
echo today\'s illumination level = $illumlevel and \
  yesterday was $yillumlevel
```

Running it on November 7, 2016, here’s what the script and the Moon Giant website report:

```
today's illumination level = 47 and yesterday was 37
```

Now it’s a simple test: is today’s level greater or less than yesterday’s level?

But wait, “waxing” and “waning” applies only to crescent and

## WORK THE SHELL

gibbous moon phases. If the moon is new, quarter or full, neither word applies in common astronomical parlance.

Seriously, who came up with these rules? Talk about complicated! Here's how this all fits together:

```
if [ $illumlevel -gt $yillumlevel ] ; then
  # we're waxing if it's getting brighter
  waxwane="waxing"
else
  waxwane="waning"
fi
if [ $illumlevel -lt 5 ] ; then
  phasename="new"
elif [ $illumlevel -lt 45 ] ; then
  phasename="$waxwane crescent"
elif [ $illumlevel -lt 55 ] ; then
  phasename="quarter"
elif [ $illumlevel -lt 95 ] ; then
  phasename="$waxwane gibbous"
else
  phasename="full"
fi
echo "The moon is currently $phasename with \
  $illumlevel% illuminated."
```

That's just about the entire script. If I run it on November 7, 2016, the moon is 47% illuminated, which makes it a quarter moon (45%–55%), so the output is:

```
The moon is currently quarter with 47% illuminated.
```

A few days later, on November 10, the output is what you would hope:

```
The moon is currently waxing gibbous with 78% illuminated.
```

Done. Nice and easy.

### Creeping Featurism

You could do plenty of things with this script to improve it and make it more powerful and flexible. The easiest would be simply to rewrite that output line so it's less grammatically awkward:

```
echo "It's a $phasename moon that's $illumlevel% illuminated."
```

Now the output will make a bit more sense as the script reports that "It's a waning gibbous moon that's 72% illuminated."

The bigger task is to allow users to specify a date and calculate the values for that particular date (including the day prior to the date specified). I would do this using the same basic `date -v` approach, but first parse users' input and break it down into month, day and year. For simplicity's sake, constrain their input to a MM/DD/YYYY format, and there's surprisingly little involved in the task.

For huge bonus points, of course, a graphical display would be nice. But that's hard to do with a shell script, right?

### Next Month

That's it for space. Next month, I'm planning to turn back to games and explore how to write a rock, paper, scissors game. You might want to study the game first so you're ready! ■

Send comments or feedback via  
<http://www.linuxjournal.com/contact>  
or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

[RETURN TO CONTENTS](#)

# Orchestration with MCollective, Part II

Automate yourself out of a job with a few well placed MCollective commands.



KYLE RANKIN

---

Kyle Rankin is a Sr. Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

---

PREVIOUS

◀ Dave Taylor's  
Work the Shell

NEXT

Shawn Powers'  
The Open-Source  
Classroom ▶

---

**IN MY LAST ARTICLE**, I introduced how MCollective could be used for general orchestration tasks. Configuration management like Puppet and Chef can help you bootstrap a server from scratch and push new versions of configuration files, but normally, configuration management scripts run at particular times in no particular order. Orchestration comes in when you need to perform some kind of task, specifically something like a software upgrade, in a certain order and stop the upgrade if there's some kind

of problem. With orchestration software like MCollective, Ansible or even an SSH for loop, you can launch commands from a central location and have them run on specific sets of servers.

Although I favor MCollective because of its improved security model compared to the alternatives and its integration with Puppet, everything I discuss here should be something you can adapt to any decent orchestration tool.

So in this article, I expand on the previous one on MCollective and describe how you can use it to stage all of the commands you'd normally run by hand to deploy an internal software update to an application server.

I ended part one on MCollective with describing how you could use it to push an OpenSSL update to your environment and then restart nginx:

```
mco package openssl update  
mco service nginx restart
```

In this example, I ran the commands against every server in my environment; however, you'd probably want to use some kind of MCollective filter to restart nginx on only part of your infrastructure at a time. In my case, I've created a custom Puppet fact called `hagroup` and divided my servers into three different groups labeled `a`, `b` and `c`, split along fault-tolerance lines. With that custom fact in place, I can restart nginx on only one group of servers at a time:

```
mco service nginx restart -w hagroup=c
```

This approach is very useful for deploying OpenSSL updates, but hopefully those occur only a few times a year if you are lucky. What you more likely will run into as a common task ideal for orchestration is deploying your own in-house software to application servers. Although everyone does this in a slightly different way, the following pattern is pretty common. This pattern is based on the assumption that you have a redundant, fault-tolerant application and can take any individual server offline for software updates. This means you use some kind of load balancer that checks the health of your application servers and moves unhealthy servers out of rotation. In this kind of environment, a simple,

serial approach to updates might look something like this:

- Get a list of all of the servers running the application.
- Start with the first server on the list.
- Set a short maintenance window for the server in your monitoring system.
- Tell your load balancers to drain any existing sessions to this server.
- Update the list of available packages for the server.
- Stop the service on that server.
- Update the software on that server.
- Start the service on that server.
- Make sure the service started successfully.
- Perform a health check to make sure the service is healthy.
- Add the server back to the load balancer rotation.
- Repeat for the rest of the servers on the list.

If any of those steps fails, the administrator would stop the update and go investigate and fix the problem. Often if there is going to be a failure, it will be at the software-update or health-check phase, and the point of this process is to make sure that if an upgrade doesn't go well, you stop at the first server before pushing broken software to the rest of the environment.

Traditionally, administrators might perform all of the above steps manually by logging in to different servers and interacting with different web interfaces perhaps. The next step they follow generally involves wrapping a series of SSH commands that would perform these

actions into a shell script and then maintain some local configuration file that defines lists of servers.

With MCollective, the process is similar with the main difference being that MCollective doesn't need to have SSH root privileges on these machines. Instead, MCollective performs its tasks by putting a limited set of commands in a job queue that all of the servers check. The commands are restricted by what MCollective plugins you have installed on a particular server, and MCollective does a good job of sanitizing input from the plugins it includes by default.

Most of the above commands in that deploy list can be completed using the default plugins MCollective includes. I use Nagios for monitoring, and although MCollective does include a plugin that lets you perform NRPE commands (a Nagios agent that runs on each server that allows Nagios to run local commands to check disk space, RAM and so on), it doesn't include anything that could directly set a maintenance mode in Nagios.

Another missing piece in the above list of commands is the ability to interact with a load balancer. Many people might skip this step these days, as they are using something like nginx's internal load-balancing abilities and may not have an easy way to set something like a maintenance mode to drain existing connections to a host. In that case, you may just skip ahead to stopping the service and let the health check detect the failure. That approach risks dropping existing connections though, and because I use Haproxy as my load balancer, I can use its built-in command mode to set a maintenance mode on specific servers if I'm logged in to the load balancer.

Fortunately, MCollective has the ability to extend its existing set of commands with your own custom plugins to perform specific tasks. Unfortunately, writing, packaging and deploying even trivial MCollective plugins can be a bit complicated the first time you do it, and it's involved enough that it would require an article all of its own. MCollective's plugin documentation is a good place to start, and in particular, the documentation on writing plugins that use MCollective's RPC framework makes the code you have to write much more straightforward, even if you aren't familiar with Ruby.

When you write a custom MCollective plugin, you choose a new plugin

name (say, haproxy) and then define a list of commands you want to pass that new plugin (such as `disable_server` and `enable_server`). If a command needs some kind of argument passed to it, you also define those. Then, you map those commands and arguments into basic command-line commands using their RPC framework, or you can dig in to using native Ruby libraries if you are familiar with that.

I wrote a custom Nagios plugin and an Haproxy plugin that would send my custom commands to their command file and command socket, respectively. So to set a maintenance mode on `server1.example.com` for Nagios and Haproxy, I would type these commands:

```
mco rpc nagios maintenance server=server1.example.com duration=5m
mco rpc haproxy disable_server server="serverrole/server1"
```

Because I took advantage of MCollective's RPC framework, I have to type `rpc` in front of my custom commands.

Next I provide the name of my plugin, then the command I want to run, followed by any custom arguments. Then on the Nagios server side, I intercept that command and format it into a format I can write to Nagios' local command file so it can execute. In the case of the Haproxy plugin, this command goes out to any server that happens to be running Haproxy. If a particular Haproxy server doesn't have my server defined in its configuration, it doesn't do anything harmful, and otherwise, it sets it to maintenance mode.

With these plugins in place, you can replace the above generic list of steps to specific MCollective commands:

- `mco find -S "domain=example.com and resource('Package[myapp]').managed=true"`
- `mco rpc nagios maintenance server=myapp1.example.com duration=5m`
- `mco rpc haproxy disable_server server="myapp/myapp1"`
- `mco rpc package apt_update -I myapp1.example.com`



- `mco service myapp stop -I myapp1.example.com`
- `mco package myapp update -I -I myapp1.example.com`
- `mco service myapp start -I myapp1.example.com`
- `mco service myapp status -I myapp1.example.com`
- `mco nrpe check_app_health -I myapp1.example.com`
- `mco rpc haproxy enable_server server="myapp/myapp1"`

I've ended up wrapping all of these commands inside a basic shell script that takes the name of a particular application as an argument, then performs the first `mco find` command to get the list of servers that have that package installed. Then at that point, I just run the next set of commands in a basic `for` loop. Where appropriate, I added a `sleep` command here and there to give a service time to come up. If any of the commands fail, the script exits out and reports the error so the administrator can investigate. Otherwise, it runs through each server in order.

Of course, later versions of this script have become a bit more sophisticated, so it can accept some custom arguments, log the output to a known log file and be more efficient in how it sleeps between commands. But the end result for the sysadmin is a simple "deployapp" script they can run that properly updates the application the right way, every time, with no risk of skipping or forgetting a server or steps in the process. ■

Send comments or feedback via  
<http://www.linuxjournal.com/contact>  
or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

[RETURN TO CONTENTS](#)

# My Love Affair with Synology

What has 48TB and is full of penguins?



**SHAWN  
POWERS**

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for [LinuxJournal.com](http://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 email at [shawn@linuxjournal.com](mailto:shawn@linuxjournal.com). Or, swing by the [#linuxjournal](https://freenode.net) IRC channel on [Freenode.net](https://freenode.net).

---

PREVIOUS

◀ Kyle Rankin's  
Hack and /

NEXT

New Products ▶

---

**IN MY RECENT "HODGE PODGE" ARTICLE** (in the October 2016 issue), I mentioned how much I love the Synology NAS I have in my server closet (Figure 1). I got quite a few email messages from people—some wanting more information, some scolding me for not rolling my own NAS, and some asking me what on earth I need with that much storage. Oddly, the Linux-running Synology NAS has become one of my main server machines, and it does far more than just store data. Because so many people wanted more information, I figured I'd share some of the cool things I do with my Synology.

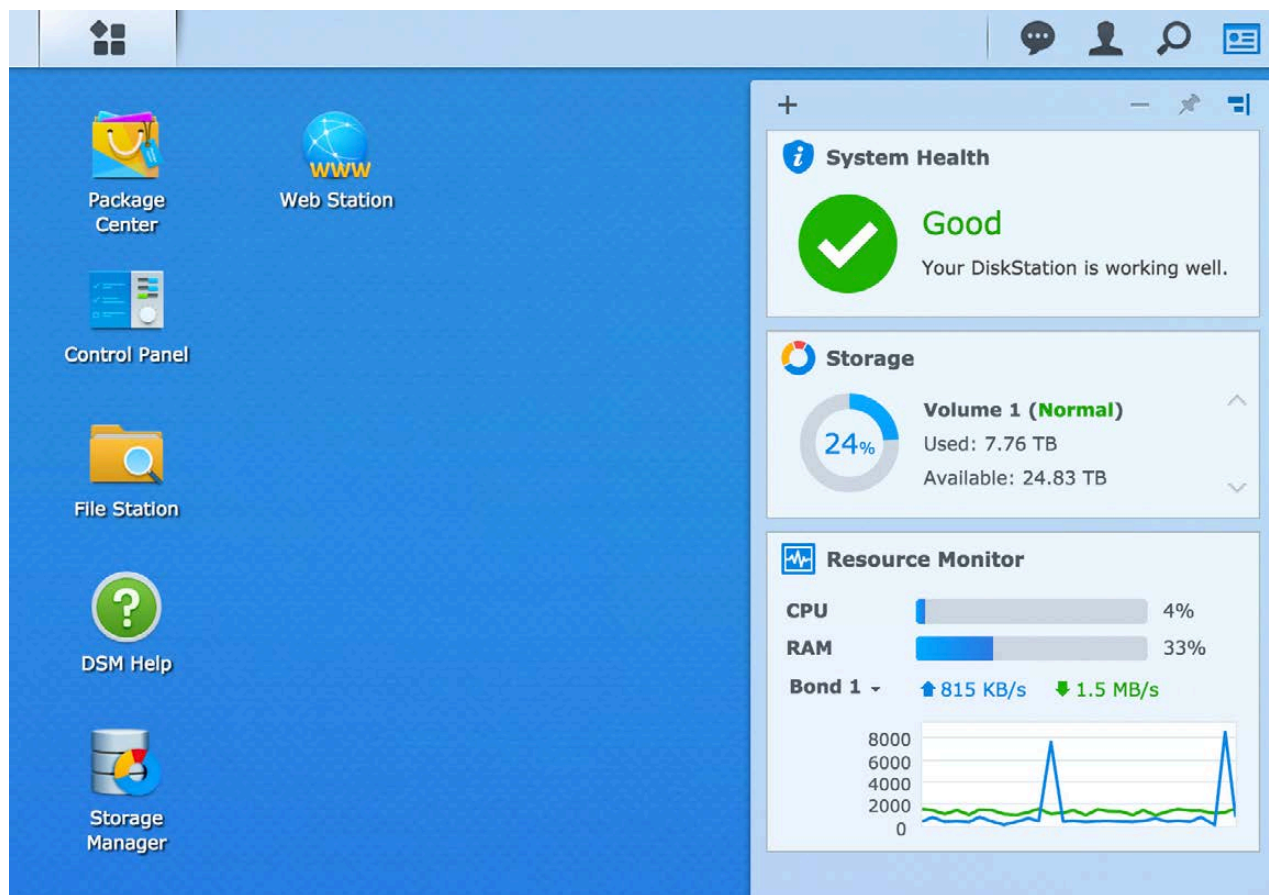


Figure 1. The Synology DS1815+ is what I use, but the entire line of Synology NAS devices shares a common interface.

### Why So Much Storage?!

I guess I should address the reason I have 48TB (36 usable) of storage (Figure 2). I store a lot of data (har har har). Seriously though, I have a local copy of close to 100,000 photos, 1000s of hours of home videos and several complete Linux distribution repositories. That takes a lot of storage! The bulk of my needs, however, comes from entertainment media. Ever since my kids first used DVDs to skate across the kitchen floor, I've been backing up my movies digitally to my server. Through the years, that has migrated from DVD ripping to Blu-ray ripping, but years of movies really add up. Even those aren't the bulk of my data, however.

I collect television series. Sometimes those collections are ripped from my TiVo, manually edited and converted to MKV. If I'm being honest, however, most of my television shows are just downloaded from torrent sites. Yes, I know it's not kosher to download torrents of television shows. But I also know that I pay more than \$200/month to the cable company for every channel available, and if I wanted to take the time, I could do the TiVo rip/edit/convert dance. I just don't have the time. Because I pay for cable access, it doesn't bother me to download television shows. (We actually do buy all our Blu-ray movies though. I'm not a proponent of



**Figure 2.** The dashboard shows you information on your NAS at a glance. I'm slowly building my collection after the horrible data loss I suffered a few years ago.

pirating things you don't have rights to.) It's okay if you disagree with my choice to download television shows via torrents, I get it. Really, I do. Just ignore those parts of this article!

## What Kind of Drives?

Don't skimp on hard drives. That's generally good advice regardless of the situation, but with NAS devices, please spend the extra money to get drives rated for NAS. I have eight 6TB Western Digital Red NAS drives. When I bought them, the WD Red Pro drives weren't available. Still, the standard Red drives are rated for up to eight drive bays, so I'm still within spec.

I haven't always been so picky about drives. In fact, I just used to get the biggest, cheapest drives I could. Since I use RAID6, a drive or two failing isn't a big deal—except that I actually had three drives fail

at exactly the same time, and I lost all my data, including family home movies that I didn't have backed up anywhere. It still hurts. So really, don't skimp on drives, it's just not worth it. (Also remember to back up, even large files. RAID isn't a backup, trust me.)

## Why Synology?

I've had Drobo's, QNAPs and multiple Netgear devices. They all sucked. No, really. The performance on every single device I've had in the past has been horrible (even with good drives), and I've never been able to determine exactly why. Once more than one simultaneous read happens over the network, they all just crap out. With the Synology, I can have four 1080p video streams going at once without any slowdown at all.

The other thing I like about the Synology is its software. Most other NAS devices have apps that you can install on the Linux system, but the Synology apps seem to be more elegant and work reliably (Figure 3). In fact, there are some incredible things I do with the NAS device that I'm sure weren't exactly what it was designed to do (more about that in a bit).

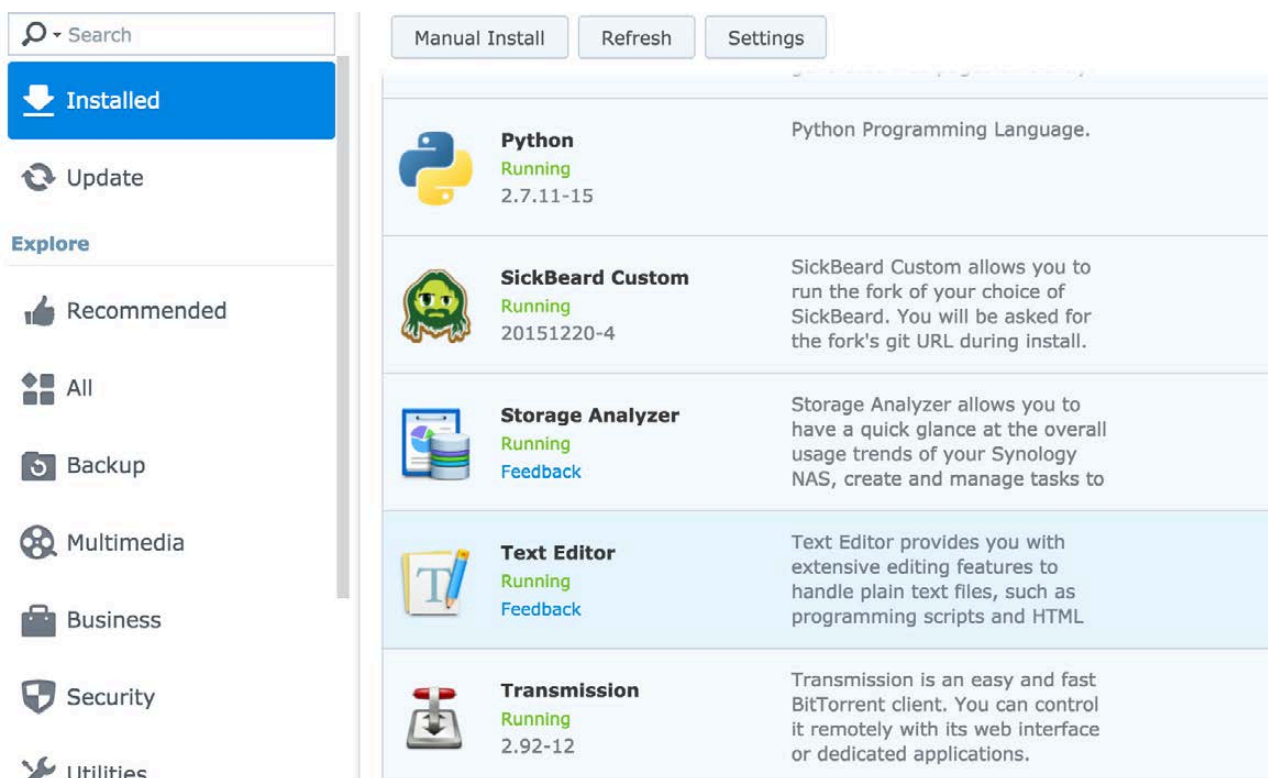


Figure 3. The apps are plentiful, and there are community-supported unofficial apps as well.

Ultimately, the biggest draw for me is how well Synology keeps itself updated and maintains its drives. It automatically does scans and integrity checks, plus it does system updates without disrupting the servers I have connected to it via NFS. Every other NAS I've used stays at whatever software version it comes with, because upgrading the firmware almost always means drive failures and server lockups. I'm sure there are procedures for QNAP and such that make upgrading possible, but the Synology does it automatically—and I like that a lot.

### TV and Torrents

I like the SickRage program not only because it automatically searches and downloads new episodes of my television shows, but also because it organizes my existing collection. I have every episode of *Star Trek* that ever has been produced (including the animated series from the 1970s), and SickRage does an incredible job of naming and organizing those files. As long as I spent ripping the *Star Trek the Next Generation* DVDs, I don't ever want to have to figure out which episode is which again!

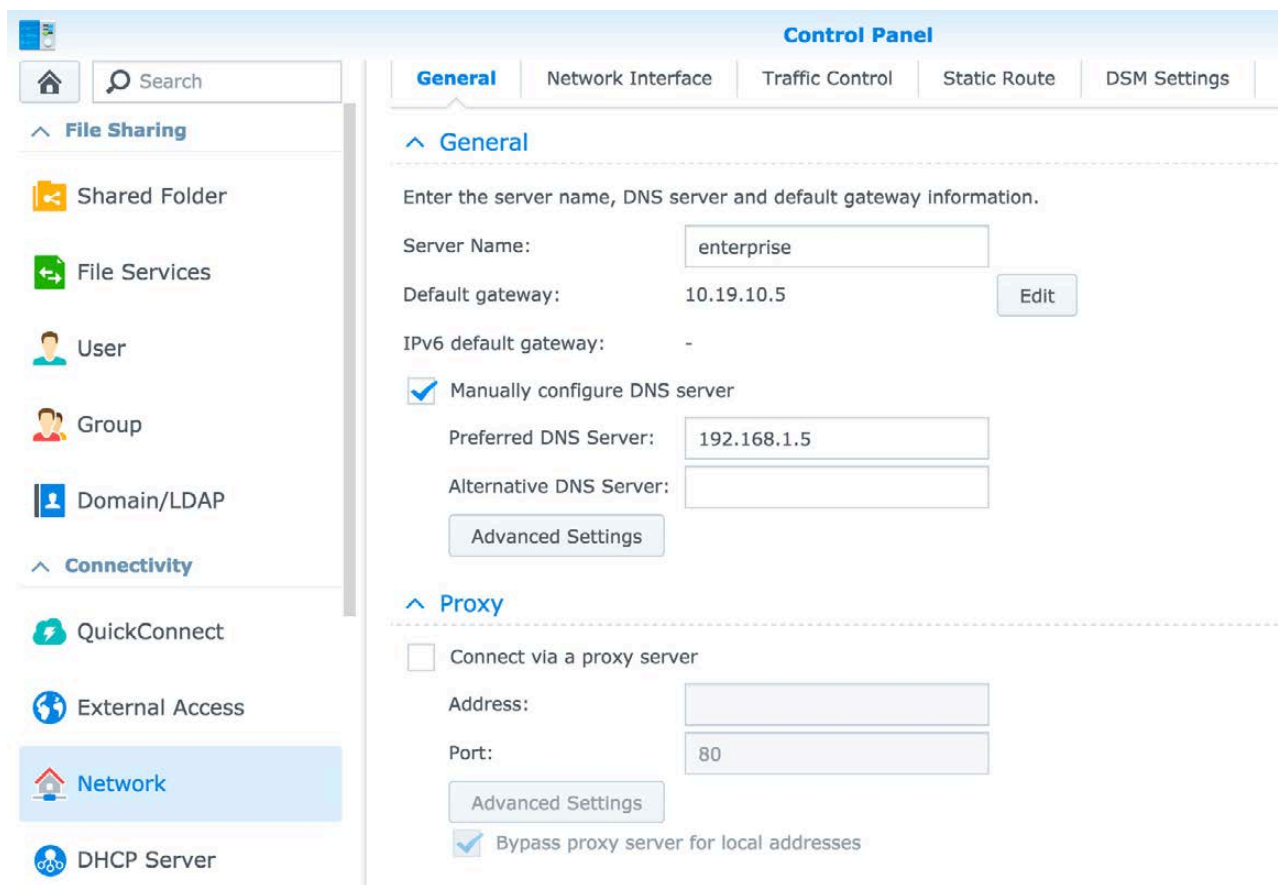
In order to install SickRage, you actually need to install "Sick Beard Custom" and then paste in the SickRage Git URL. The short version of the story is that Sick Beard was the original program, but the developer stopped developing it, so folks forked it, and SickRage is the best fork out there, by far. Even if you're not using Synology, you should be running SickRage. Head over to <https://github.com/SickRage/SickRage> for the repo or <http://sickrage.github.io> for the home page.

SickRage supports lots of torrent clients, and it supports NZB too. I've found NZB to be less reliable than it used to be, so I've moved back to 100% torrents. I like the Transmission web interface, so that's what I use on Synology. It's another maintained app, so just search for "transmission" in the package installer application. Integrating Transmission and SickRage is beyond the scope of this article, but rest assured, it's not difficult. SickRage is designed to work with Transmission, so setting it up is easy. *Warning:* if you use SickRage and Transmission to download television shows, you *will* get DMCA take-down notices from your ISP. Apparently the production companies disagree with my rationale for downloading TV episodes. Thankfully, I have a solution for that.

## Networking and Traffic Routing

My Synology device has four Gigabit Ethernet ports. I think that's overkill, but since the software allows me to bond the four ports together (even with a switch that doesn't support 802.3ad), I'm happy to have more bandwidth than I need. I never have an issue with throughput, even when streaming those multiple video files mentioned above.

Since Synology supports VPN connections, the first thing I did was set up my privateinternetaccess.com account so my torrents would be directed through the VPN. I haven't gotten port forwarding to work through the VPN, but even without a redirected port, my torrents download fine. The problem is my VPN connection occasionally goes down. When it does, the torrents go through my gateway, and even when the VPN comes back up, the tracker connects me via the non-VPN connection. And, I get DMCA notices. This is very frustrating. So I decided to remove the gateway device



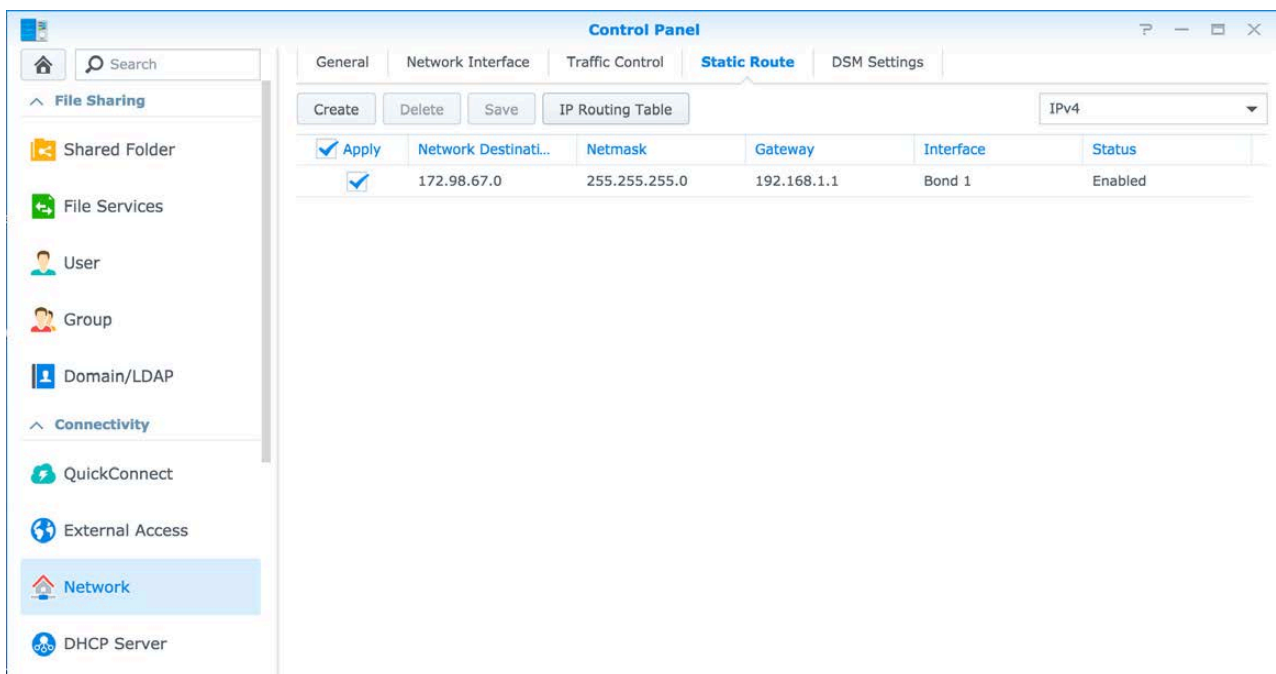
**Figure 4.** Notice the gateway is in the 10.x.x.x range, which is not what I use on my local network. That is assigned by the VPN.

# THE OPEN-SOURCE CLASSROOM

from the Synology altogether! Bear with me.

I have a network address assigned on my local network so LAN computers can connect. That works fine. Without a gateway specified, however, the NAS can't connect to the internet for torrents, SickRage or even system updates. But when the VPN is connected, it sets the gateway address automatically to an address on the other side of the VPN (Figure 4). As long as my VPN is connected, the system has a gateway assigned, and it can access everything through the VPN. If the VPN goes down briefly, rather than defaulting to the local network gateway, it just can't connect to the internet. Once the VPN is re-established, it reassigns a VPN gateway, and boom, the NAS is back online! The only problem is how can I connect to the VPN if I can't get on the internet? The answer: static routes.

If you look at Figure 5, you'll see that I have a static route set up so that traffic going to the IP address of my VPN goes through my LAN's gateway. Since it's only a static route for that network, the rest of the internet is still inaccessible. I also could do fancy firewall work and allow the NAS to access only the VPN and drop all other packets, but I like the solution to be self-contained. That way, if I change routers or router configs, I don't have to worry about getting DMCA notices.



**Figure 5.** This is the sneaky static route so I can connect to my VPN, but nothing else.



# THE OPEN-SOURCE CLASSROOM

The Synology also will act as a router, forwarding traffic. That means I can point my Roku to the Synology as its gateway device, and I'm able to watch local blackout games on the MLB.tv app, because all the traffic goes through the VPN. The only change I have to make is on my DHCP server, which gives the Synology's IP address as the Roku's gateway address. It works perfectly and saves me setting up another VPN to get around MLB's regional restrictions. (Honestly, I usually watch baseball games on TiVo, but occasionally the game is on only via streaming, and I like having that option.)

## Backups

Remember when I said RAID wasn't a backup? Yeah, I meant that. I've lost too much valuable data through the years to depend on RAID to protect my files—even when the drives and NAS device seem to be

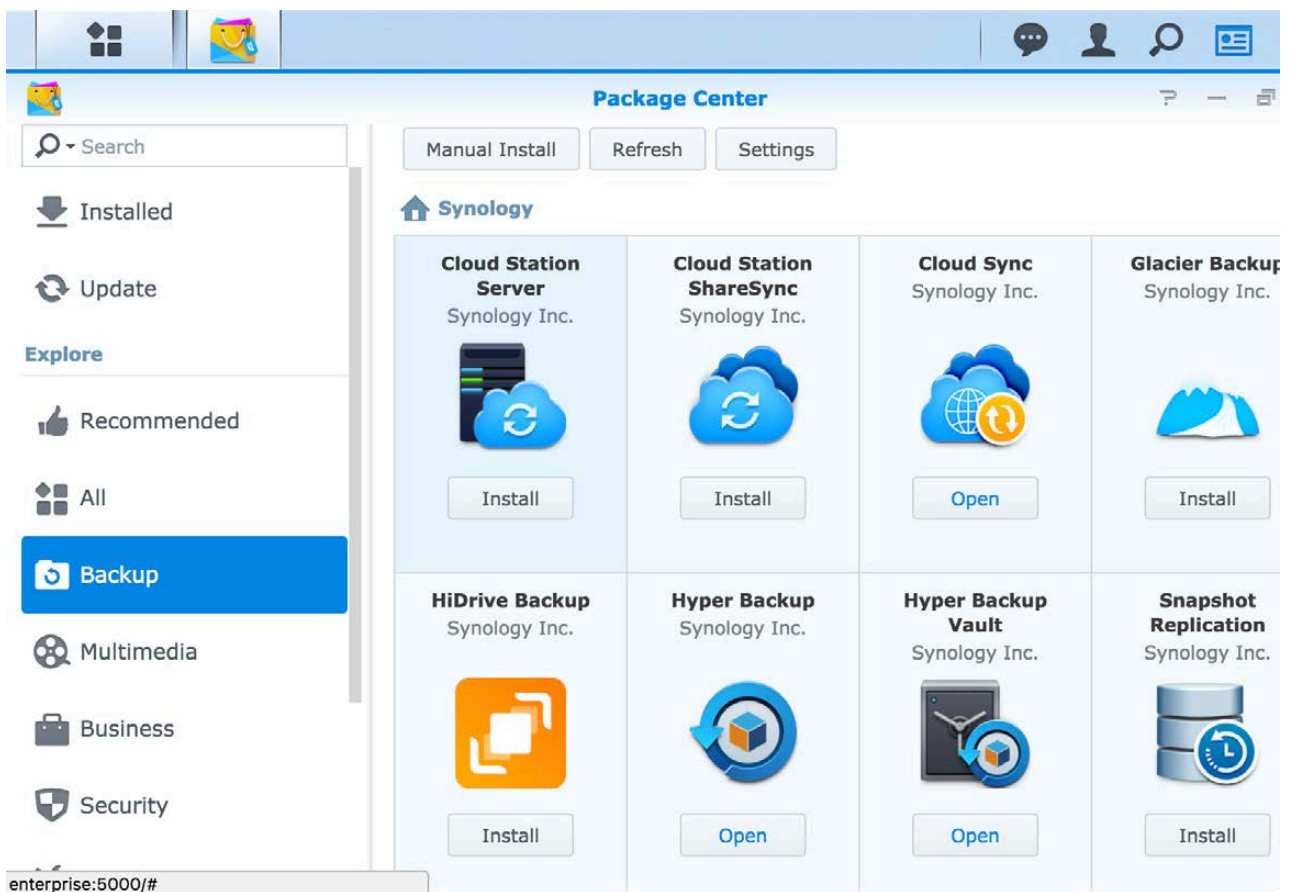


Figure 6. Backup solutions are in great supply.

more solid than any I've had in the past. Thankfully, Synology has a few different backup options (Figure 6). The most practical one for large amounts of data is the Hyper Backup app. It has the ability to copy your entire NAS to a variety of destinations. Whether you choose to buy another Synology NAS and store it in your shed or back up your data to Amazon Glacier, the same Hyper Backup program can handle the regular updates.

I don't want to pay for Amazon Storage, even though the Amazon Drive Unlimited is decently priced at \$60/year. I worry that my 30TB would cause Amazon to invent a reason to suspend my account. Plus, it would take so long to back up my entire data store to the cloud, that it literally might never get done. Right now, I just back up my irreplaceable files (home movies, photos and so on). Someday I hope to get a second Synology NAS and set up that "mirror in the shed". Still, Synology has so many backup options, it's hard to find a reason to delay setting up a backup solution!

### Things I Don't Do

The Synology had a decent processor, and the RAM is even upgradeable. Still, it's not a beefy server when it comes to resource-hungry applications. For example, even though the Plex Media Server is available in the package management system, I'd never install it. Plex uses way too much CPU to transcode video streams. I'm thankful the Synology is powerful enough to stream the actual video files over fileshares, but the thought of transcoding 1080p MKV streams in real time? It's a bad idea. I have a standalone server I use for Plex Media Server, and while it can transcode at least four full resolution video streams, it's also a huge i7 CPU with a boatload of RAM. Unless you're doing minimal streaming with low-resolution video, I encourage you to avoid Plex Media Server on any NAS device.

I also can't run the really amazing reverse proxy server on Synology. The setup is easy, and the configuration is very intuitive, but my VPN/no-gateway setup means that the reverse proxy doesn't work outside my network. Even if I forward a port to the NAS from my router, it tries to send responses out the VPN connection and fails. Reverse proxies are easy enough to configure on any other machine

in my network, so it's not a huge loss, but it's worth noting that it's something my crazy VPN system breaks.

### **Not the Only Option**

Before you think I was paid by the folks at Synology to brag about their product, I will freely admit that a big tower server with a bunch of hard drives and software RAID makes for an incredible NAS. It means you can beef up the hardware too and do things like run Plex Media Server. I simply like the efficiency of the Synology devices. They're fast, cool running and just sip electricity. I'm sure there are other brands of NASes that do a decent job too, and Synology isn't perfect. In all honesty, however, it's the best product I've been able to find, and I have literal piles of junk NAS devices that just couldn't do the job. If you're looking for a NAS device, in my opinion, you can't go wrong with Synology. ■

Send comments or feedback via  
<http://www.linuxjournal.com/contact>  
or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

**RETURN TO CONTENTS**

# NEW PRODUCTS

---

## PREVIOUS



Shawn Powers'  
The Open-Source  
Classroom

## NEXT



Feature: Online Privacy  
and Security Using a  
Password Manager

---



## Rogue Wave Software's TotalView for HPC and CodeDynamics

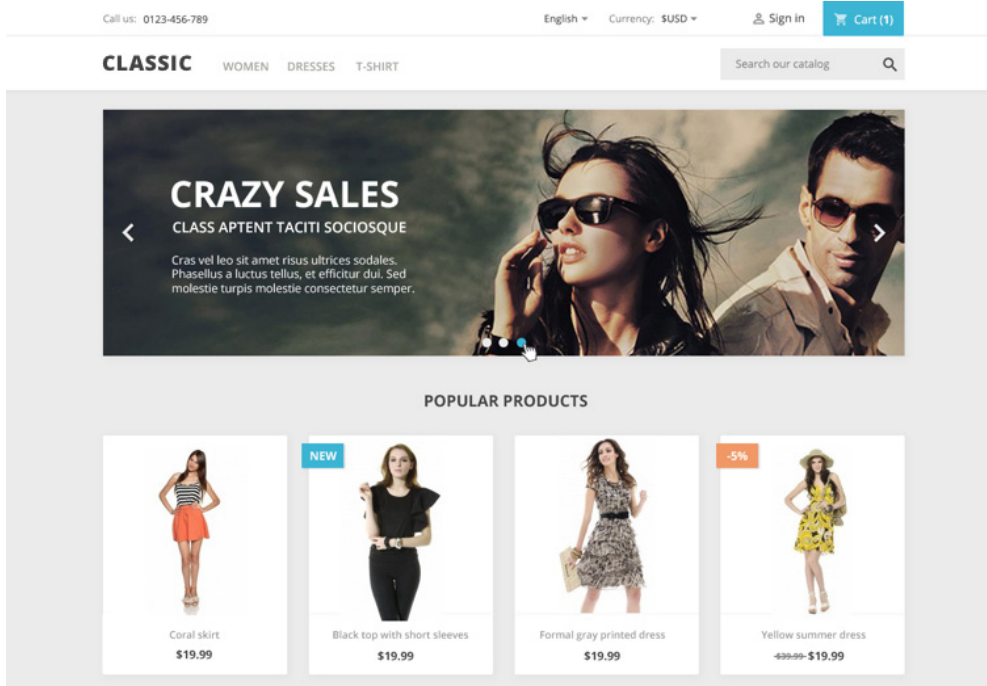
New versions of not just one but two dynamic analysis tools from Rogue Wave Software were unveiled recently to pleased developers everywhere. Upgraded TotalView for HPC and CodeDynamics, versions 2016.07, improve the diagnosis and correction of bugs, memory issues and crashes at execution. The heart of these tools' innovation is support for 64-bit ARM v8-A architecture. ARM's presence continues to expand the ecosystem for the HPC space and enables highly integrated, energy-efficient solutions. With TotalView for HPC and CodeDynamics, customers using ARM platforms have a robust, scalable dynamic analysis solution for their complex HPC clusters and codes. Also included are overall enhancements, platform updates and improved support for operating systems and compilers.

<http://roguewave.com>



# Panther MPC, Inc.'s Panther Alpha

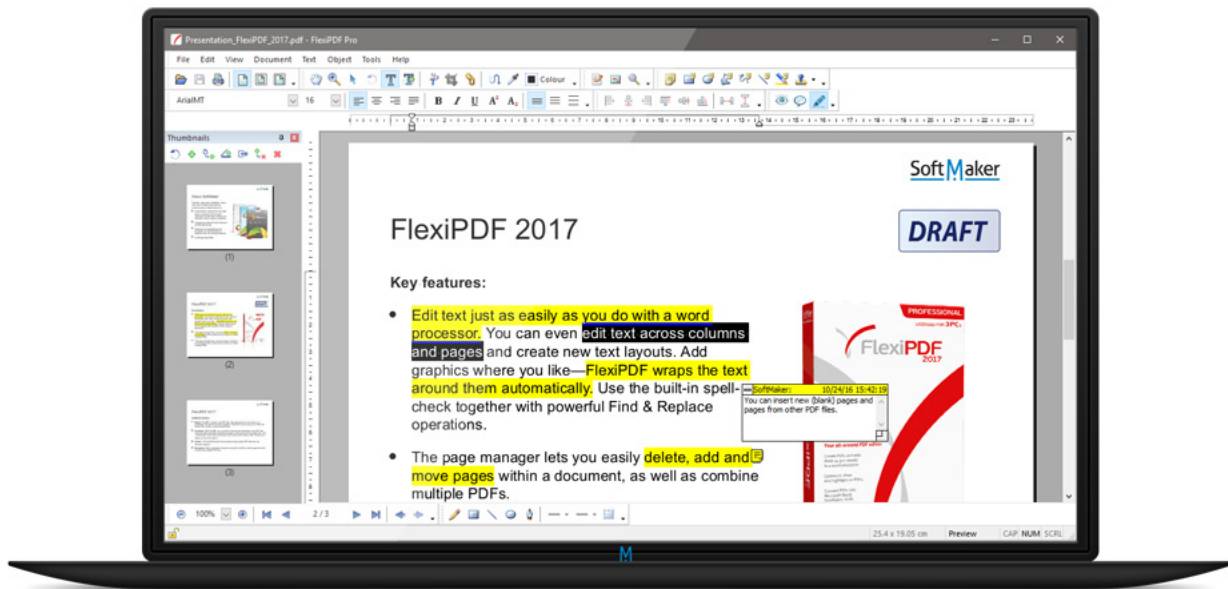
Functional, powerful and there when you need it, unobtrusive when you don't—that's how Panther MPC, Inc., sums up the company's new Panther Alpha personal micro PC that features the company's powerful, easy-to-use Linux-based Panther OS. Panther Alpha combines full desktop functionality with an ultra-customizable Linux OS that fits in the palm of your hand. Panther says its new device could be possible only now thanks to a culmination of years of industry innovation and development, namely the power of today's ARM chips and an improved emphasis of Linux on ARM. The Panther Alpha is whisper quiet, with no fan or HDD noises, and is powered by an energy-efficient ARM quad-core chipset. Unlike most micro PCs, asserts Panther, the Panther Alpha offers a true desktop experience with mouse, keyboard input, microSD card slot, Ethernet, Wi-Fi, Bluetooth and USB functionality. The portable and compact shape allows it to integrate seamlessly into any environment—business, home or school. "The only PC you'll ever need", adds the company. <http://panther-mpc.com>



## PrestaShop

Helping people overcome the challenges of building and growing an online business is what the PrestaShop open-source ecommerce platform is all about. The significant PrestaShop 1.7 release provides innovations focused on three themes: sell faster, create easier and code better. PrestaShop 1.7 users will sell faster due to the entirely redesigned UX that helps merchants with the daily management of their stores via more effective back-office management and a drastically reduced time to place products online. PrestaShop 1.7 users will create easier with new features like the starter theme, a simple and flexible tool with all the functionality needed to design the perfect store. The starter theme gives web designers significant opportunity for creativity and customization. Finally, PrestaShop 1.7 users will code better with the incorporation of new tools and standards, including the popular Symfony framework and PHP 7 compatibility. The goal is to improve site performance and code security as well as to facilitate better development. PrestaShop 1.7 is already available in 25 languages and through community crowdsourcing, it gradually will be available in many more.

<http://prestashop.com>



## SoftMaker's FlexiPDF

Editing PDFs is now as easy as word processing. This is SoftMaker's promise thanks to its new FlexiPDF 2017, a new PDF editor that "masters the creation of new PDF files as well as the editing of text, graphics and drawings in existing ones". Available in Standard and Professional versions, FlexiPDF goes far beyond basic editing functions, empowering users to change almost any aspect of PDF files. Just like in standard word processors, FlexiPDF 2017 offers an integrated track changes function, built-in spell checker, commenting function, search and replace, highlighting and exporting of PDFs to TextMaker, HTML, RTF, Microsoft Word and EPUB formats. The integrated OCR function in FlexiPDF converts scans into editable PDF files. FlexiPDF comes with its own PDF printer driver that lets users create high-quality PDFs directly from within any Windows application that can print. FlexiPDF Professional is perfect for the translation of PDF files into other languages. This is because it extracts the text of a PDF document in a format that can be opened by common translation programs. It then imports the translation back to the original PDF document—in exactly the right spot.

<http://flexipdf.com>

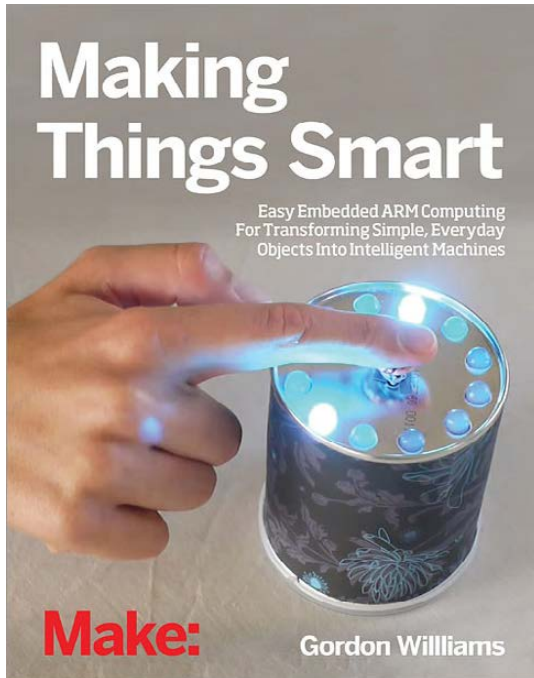


## Goldtouch Semi-Vertical Mouse

“A brilliant combination of ergonomics, comfort and performance” thunders Goldtouch, a leader in desktop ergonomics, about its new Semi-Vertical Mouse. “Perfectly sloped at 66 degrees” to prevent wrist stress and provide the ultimate comfort fit for healthy computing, the Semi-Vertical Mouse is Goldtouch’s newest addition to its mouse library. The new mouse provides a happy medium between the traditional ergonomic mouse (sloped at 25–30°) and the traditional vertical mouse (sloped between 85–90°). The in-between slope guides users into a comfortable grip, provides a nice resting position for the hand and enhances the user experience through great mousing precision. Goldtouch says that the Semi-Vertical Mouse expands its already wide variety of ergonomic solutions to fit every type of user and user preference.

<http://goldtouch.com>





## Gordon H. Williams' *Making Things Smart* (Maker Media, Inc.)

Pretty much anything in the O'Reilly spin-off Make: series is like catnip to us Linux cats, and the new book *Making Things Smart* is no exception. The book is subtitled *Easy Embedded ARM Programming*

*For Transforming Everyday Objects Into Intelligent Machines* and is authored by Gordon H. Williams. The book is Williams' tool for teaching readers the fundamentals of the powerful ARM microcontroller by walking beginners and experienced users alike through easily assembled projects composed of inexpensive, hardware-store parts. In rebellion to the many current ARM programming books that take a bland, textbook-ish approach with focus on complex, beginner-unfriendly languages—think C or ARM Assembler—*Making Things Smart* utilizes Espruino (JavaScript for Hardware) to flatten the learning curve.

<http://oreilly.com>



## Buddy Platform Limited's Parse on Buddy Service

With Facebook's Parse mobile back end as a service shutting down in January 2017, developers are in a bind. The vise is squeezing tighter since the open-source Parse Server product released by Facebook, asserts Buddy Platform Limited, was not designed to support high volume, commercial-grade apps from organizations seeking the breadth of the original platform. To solve this issue and facilitate an easy migration path to a robust alternative, Buddy launched an enterprise-grade Parse service offering called Parse on Buddy. Parse on Buddy's developer says that its solution is closest to the original Parse service, including many features not part of the Parse Server codebase released to the Open Source community (for example, cloud code, multiple apps per account support, PPNS, file storage and more). Parse on Buddy was designed to be as familiar to developers as possible, right down to using the same pricing model. It also features an easy three-step migration process for apps still on the original Parse service that requires only a small change to a mobile app's codebase and no remapping of APIs or other new code.

<http://buddy.com>



## VMware's Clarity Design System

By combining user experience (UX) guidelines and patterns with the front-end code in one solution, VMware's Clarity Design System represents a new concept in the design systems space. Clarity Design System is VMware's open-source project—"designed by designers and built by developers"—that unites UX, HTML/CSS and Angular for creating cohesive experiences for any application. VMware began building Clarity based on readily apparent needs for an internal design system that unified VMware's portfolio of products and accelerated product development through reusable components. Building alongside product teams, the Clarity team was able to tackle the complexities of enterprise use cases while striving for simplicity familiar to consumer applications. Now dozens of VMware product teams have adopted Clarity internally in their work. <http://vmware.github.io/clarity>

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

[RETURN TO CONTENTS](#)

# ONLINE PRIVACY AND SECURITY USING A **PASSWORD MANAGER**



Password managers make it easy to have unique user names, unique email addresses and unique passwords for each account. They also provide a secure store for extra account details. Additional features like notes and attachments allow you to use password managers as secure escrow files like digital safety deposit boxes.

**der.hans**



PREVIOUS  
New Products

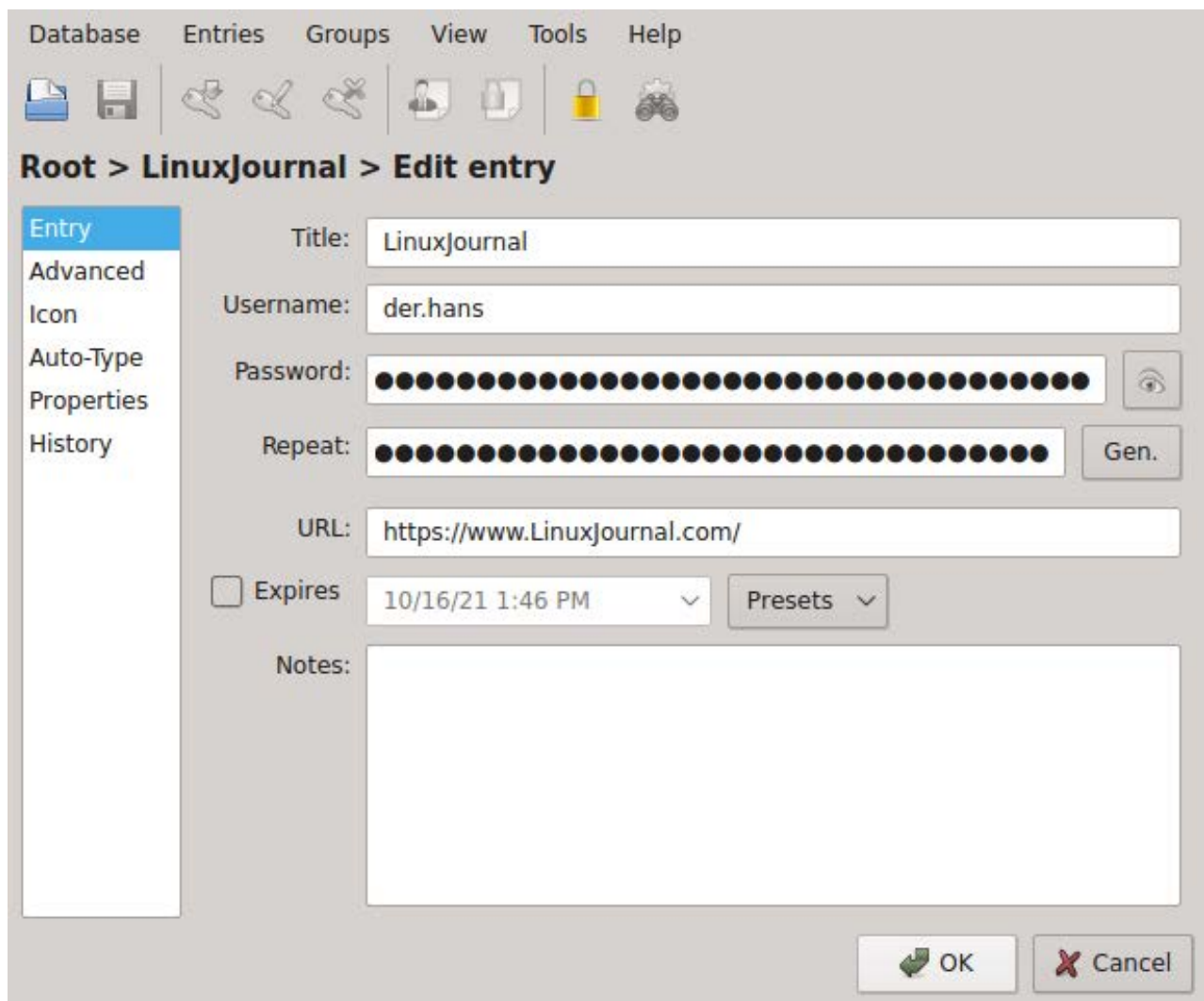
NEXT  
Feature: Low Power  
Wireless: CoAP



**I**nternet and cloud companies do not have perfect security. You're all familiar with large-scale data theft from big corporations. Although most enterprises do a great job considering the attacks against them, you can do your part by protecting your own accounts.

It's common wisdom to use a different password for each account. Better yet is using a unique email address as well. Doing so quickly becomes unwieldy, however, so you find you need an external brain.

Password managers function well as this external brain and can help with more than just passwords. They make it easy to have unique entries for user names, email addresses, security-question answers and much more.



The screenshot shows a password manager interface with a menu bar (Database, Entries, Groups, View, Tools, Help) and a toolbar with icons for file operations and security. The main window is titled 'Root > LinuxJournal > Edit entry'. On the left is a sidebar with options: Entry (selected), Advanced, Icon, Auto-Type, Properties, and History. The main area contains the following fields and controls:

- Title: LinuxJournal
- Username: der.hans
- Password: A field of 20 black dots with a visibility toggle icon on the right.
- Repeat: A field of 20 black dots with a 'Gen.' button on the right.
- URL: https://www.LinuxJournal.com/
- Expires: An unchecked checkbox, a date/time field '10/16/21 1:46 PM', and a 'Presets' dropdown menu.
- Notes: A large empty text area.

At the bottom right are 'OK' and 'Cancel' buttons.

Figure 1. Password managers should have fields for record name, user name, website and notes.

## Rather than creating a whole new email account, you can likely take advantage of sub-addressing.

Password managers securely encrypt data before storing it. In addition to passwords, they should have fields for record name, user name, website and notes.

The following example uses the `apg` (Automated Password Generator) command to create a random string of text:

```
$ apg -n1 -Mn1  
vucapob7
```

Now you have a unique user name for your bank. The next social-media site thieves will have to figure out both your user name and your password to attack your bank account.

Use a unique email address. Rather than creating a whole new email account, you can likely take advantage of sub-addressing.

Some email providers have sub-addressing, which allows a separator character and then a token. When delivered, the email provider ignores the separator character and the text after it. For example, `you+vucapob7-mybank@example.com` would be delivered to `you@example.com`. See the Email Sub-addressing sidebar for more information.

Unique email addresses make it harder for thieves to social-engineer companies, as they won't have your user name, whether or not it's an email address.

Since you're storing credentials in a password manager, you don't need to memorize or even see them. In fact, now you've turned your password manager into a credential manager holding unique passwords, unique user names and unique email addresses. Just a few paragraphs in, and you're already powering up!

# CREATING GOOD PASSWORDS

When using a credential manager, you need to memorize only a few passwords—for instance, your password to authenticate to your computer and credential manager, and both for work as well.

Here are some tips on building strong, but memorable passwords:

- 1) Use multiple unrelated words, at least four.
- 2) Include numbers, capital letters (not just at the beginning of a word) and punctuation. Perhaps use them rather than spaces.
- 3) If fluent in multiple languages, use them.
- 4) Use nonsensical, uncommon words—no common phrases. For instance, start with four unrelated words. This is probably fine for computer and network logins that change every few months:

`anteater twiddle aquamarine drove`

- 5) Use more than four words for your password manager where backups might be captured for long-term, offline attacks:

`ameisenbaer twiddle aguamarina drove preamble esac`

- 6) Even better, add some non-space transitions:

`aMeisenbaer%tWiddle aGuamarina9dRovezpReamblezeSac`

On a mobile device, that last example is a pain. Especially when using a simpler password, minimize the number of entries in the mobile password database, and change anything in it on a regular basis in case some app ships the file off to a botnet for cracking or the phone is stolen.

# One key to having unique security questions is to lie.

But, as the infomercials say, there's more.

The notes field allows you to store multi-factor emergency codes, shoe size and anything else that might be useful for the account. You could store customer-service phone numbers, cheat codes for getting to an actual person and notes from the last call.

Some sites have undocumented password requirements, such as not allowing % in a password. The notes field is a good place for a reminder for you. If you use NoScript and a cookie blocker, you also might want to add notes about required JavaScript and cookie domains to use the site.

One key to having unique security questions is to lie. The company doesn't need to know your grandma's favorite first high school mascot. It just needs a response only you can give. Random text to the rescue:

```
$ apg -n1 -m15 -Mn1
icunyedgicekoco
```

That's a fun one, but it'll be challenging if you have to say it over the phone. `apg` can help with that. The `-t` option says to give pronunciation guides:

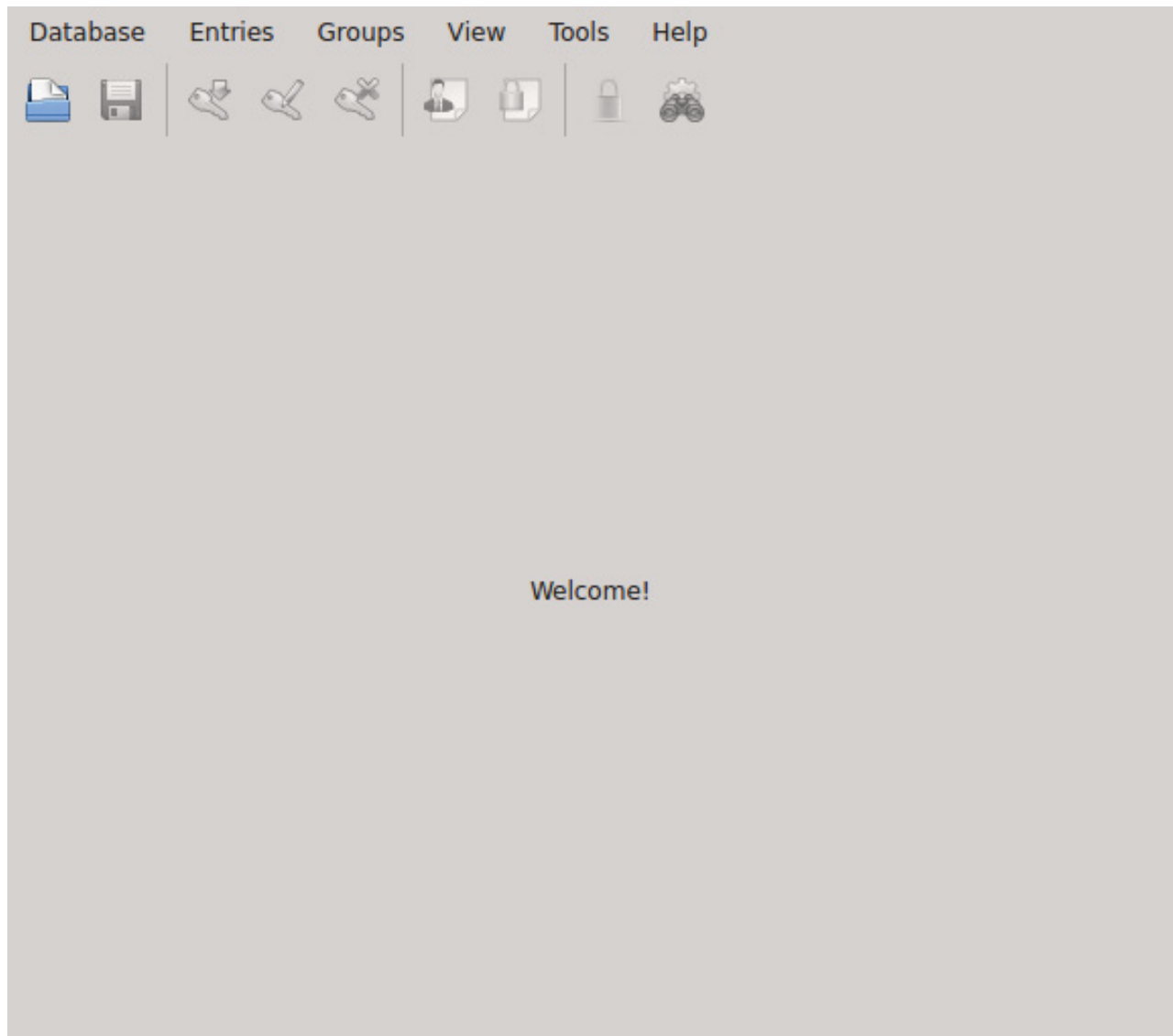
```
$ apg -n1 -m15 -Mn1 -t
icunyedgicekoco (ic-un-yed-gic-ek-oc-o)
```

It still sounds like a foreign language, but it's hardly guessable. The credential manager doesn't care—it's just data.

## KeePassX Credential Manager

I recommend using KeePassX combined with KeePassDroid. This article uses KeePassX 2.x. Most of it applies also to the KeePassX 1.x versions,





**Figure 2. KeePassX Opening Screen**

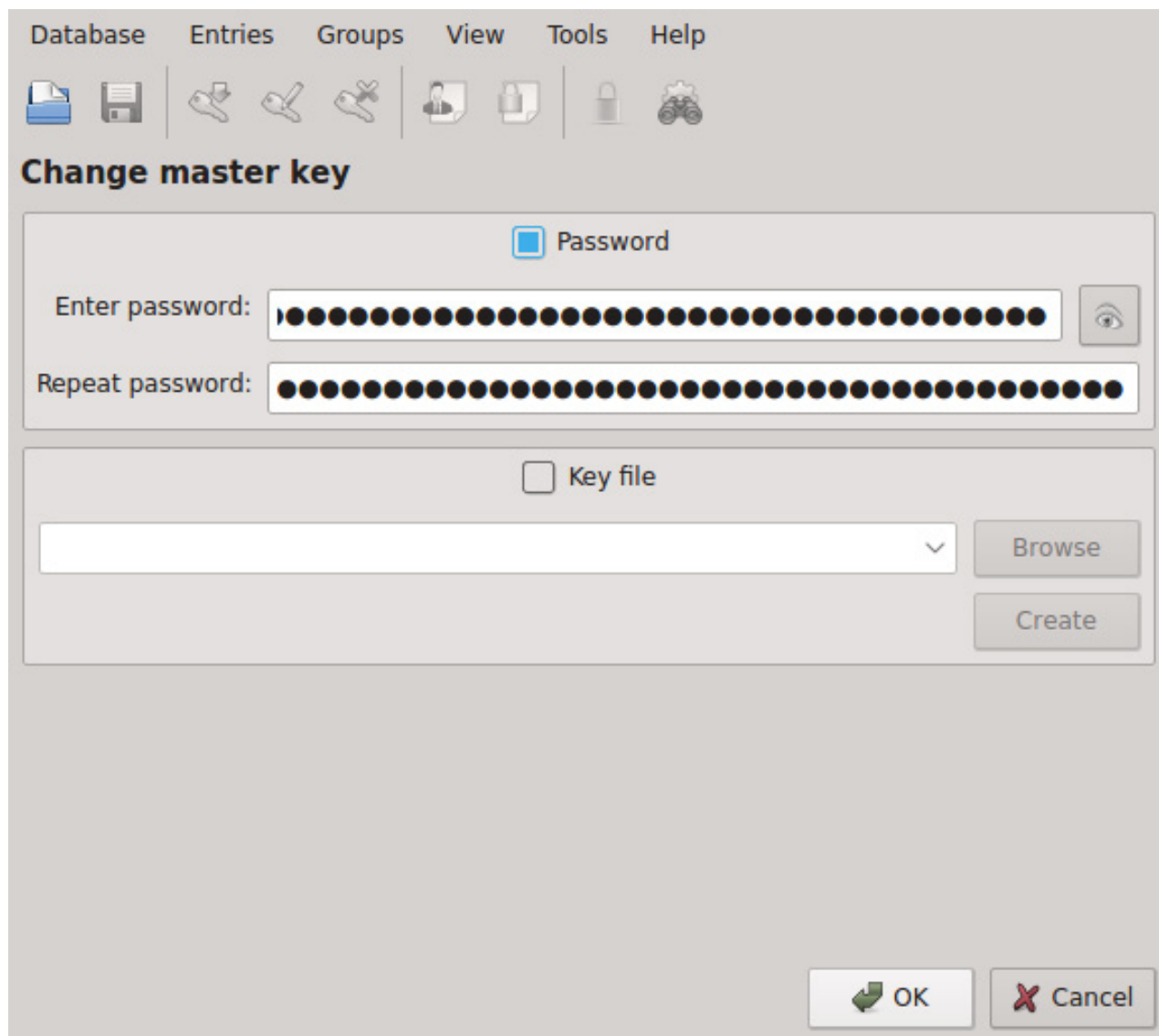
but there are some differences.

Select Database→New Database to create a new credential database.

This password you have to memorize. If you forget the password for your KeePass file, you can't open it. See the Creating Good Passwords sidebar for how to choose wisely.

You also can use a key file. The key file holds the key used to unlock the database, and your password opens the key file. An advantage is that the keyfile can be stored separately from the credential database.

Once you've entered your password, I recommend you save the file and close it, then open the file again to make sure you can type in your new

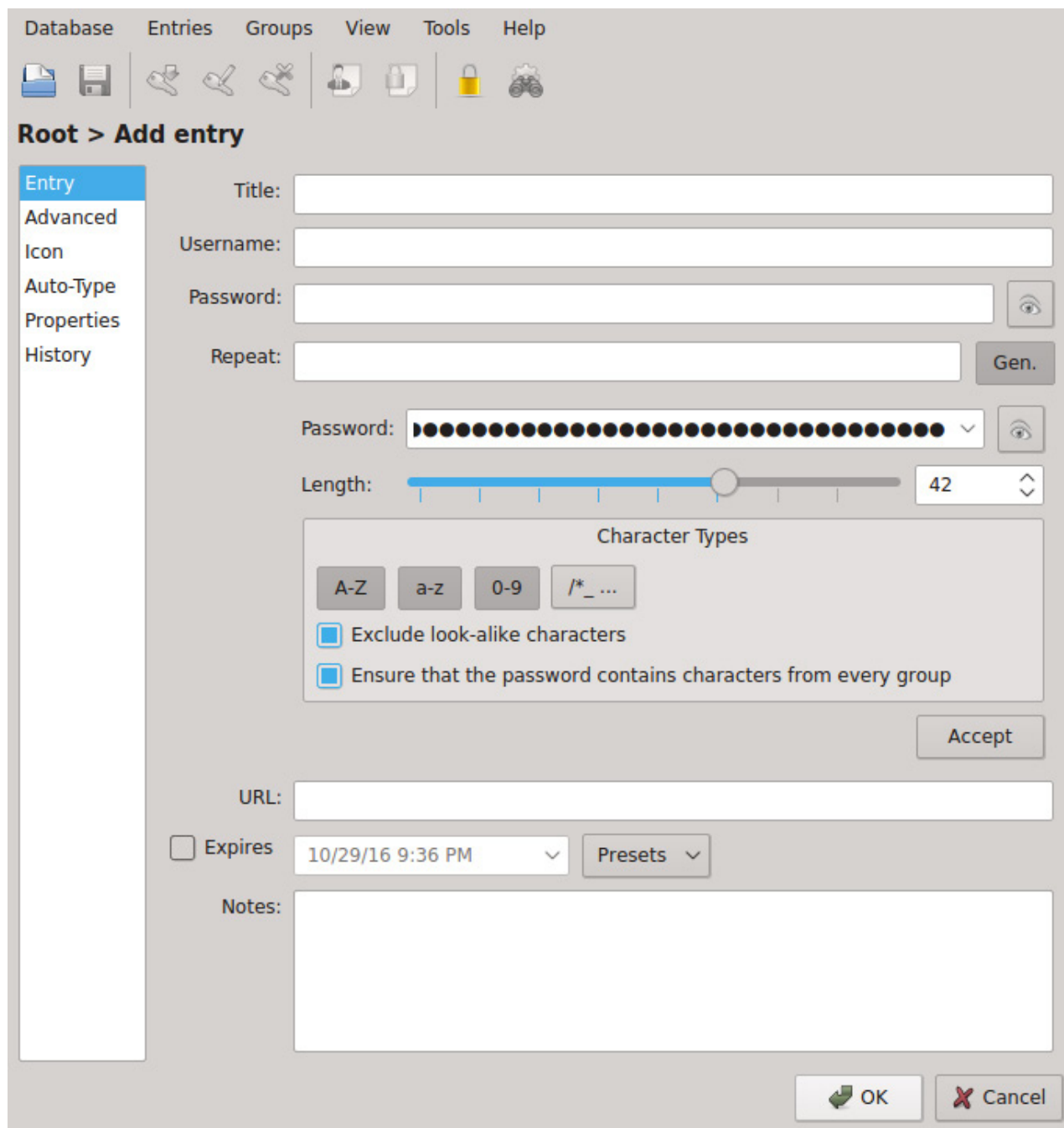


**Figure 3. Creating a New Database**

password. Open and close it three or four times to help you memorize the password. After 15 minutes, do the close and open dance again, then make sure you test it the next morning as well. Forgetting that password is the same as losing the file—the data is unavailable.

To add a new entry, either click the key icon with the green arrow or the Entries→Add New Entry drop-down menu. The drop-down shows a third option, the Ctrl-n shortcut.

KeePassX can autogenerate passwords. In addition to specifying character groups like upper and lowercase letters, numbers and special characters, you can choose length and exclude look-alike characters, such



**Figure 4. Autogenerating a Password**

as O and 0 and 1 and l.

Choose a long length like 42 characters since you're not memorizing the passwords. The eye button will let you view the random text if you need to (which is useful if a site discourages good passwords by blocking password pasting). Click the accept button to get KeePassX to store the new entry, then save the file.

In addition to the notes field, KeePassX 2.x has fields for additional attributes. The latter is a better place for storing security questions and answers. Although both fields display the values in plain text, the default screen does not show additional attributes, and there's a drop-down menu to use them.

Additional attributes can be found in the Advanced section for an entry. To copy them, use the Entries drop-down, select Copy Attribute to Clipboard, then select the attribute you want.

For securely copying a secret, it's important to keep it hidden and keep it safe. The password stays hidden because it isn't displayed. It's kept safe because it expires out of the clipboard. By default, the clipboard is cleared after ten seconds.

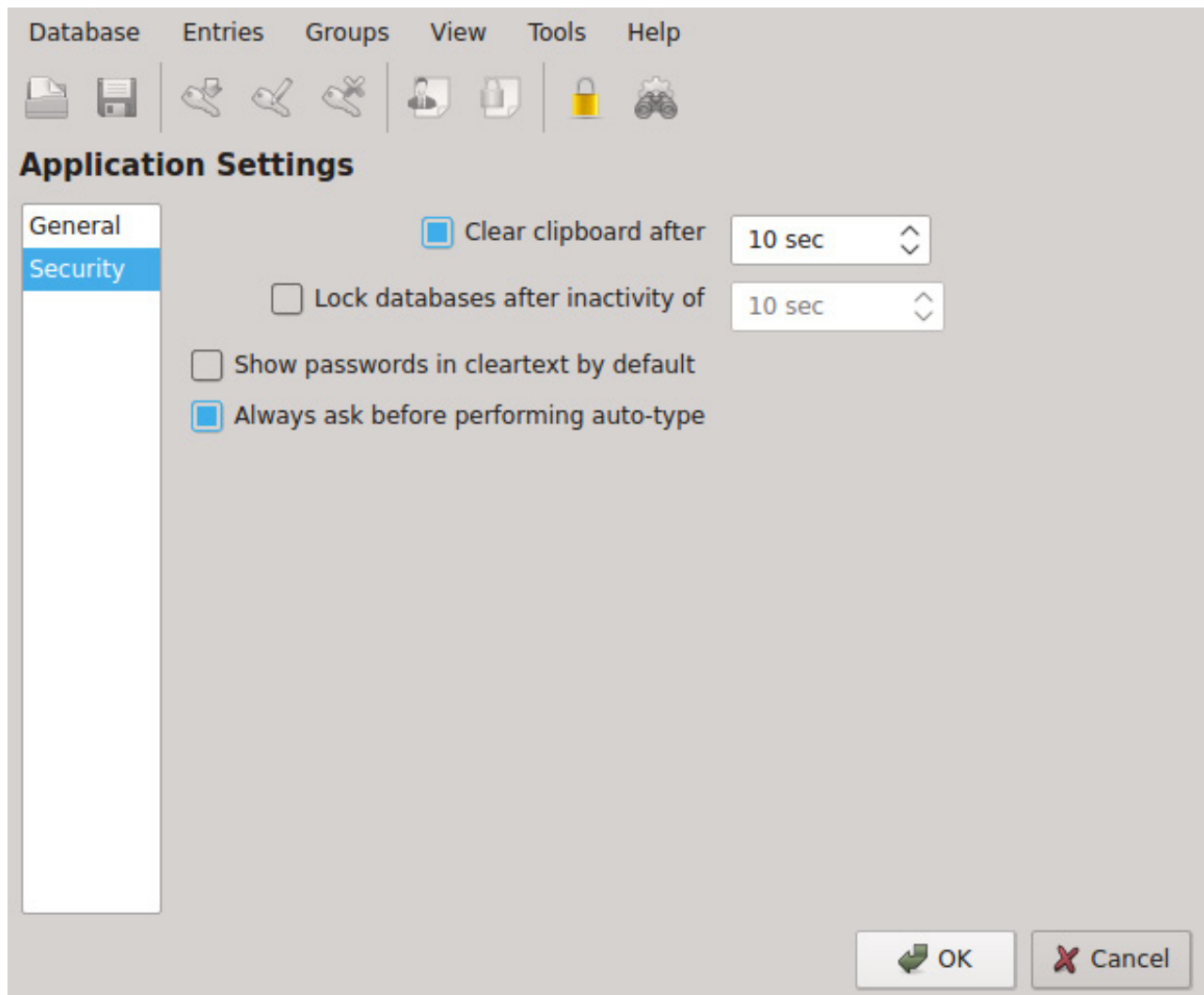


Figure 5. Clear Clipboard Setting

KeePassX provides handy keyboard shortcuts. Ctrl-b copies the user name into the clipboard for pasting, and Ctrl-c copies the password. Review the drop-down menus to find more shortcuts.

### Credential Manager Features

You should require the following features in a credential manager:

- Clipboard clearing.
- Password generation.
- Pronounceable password option.
- Encrypted data with operating-system-independent storage.
- Backups and data liberation.

The clipboard should be cleared automatically after a short time. There's no need for it to stick around and get pasted somewhere accidentally.

A credential manager should have a password generator. Even better is a random text generator available from anywhere in the application that simplifies creating random values for security questions and responses, sub-addressing tokens and birthdates.

The random text generator should have an option for pronounceable passwords. KeePassX 1.x does, but KeePassX 2.x does not. Hopefully the feature will be added back in. [apg's pronunciation guide](#) is quite useful. Neither version of KeePassX has that.

The data file must be encrypted and interoperable. In order to be trusted, the encryption software and algorithm need to be free and open so they can be audited independently. KeePassX is GPLv2 or later, and KeePassDroid is GPLv3. Both incorporate code under other free software licenses.

Interoperability means usable by multiple applications. KeePassX and KeePassDroid use free and open KeePass file formats. KeePass, kpass and other applications also can use the files. None can open the file without the master password. They are your passwords; you should

have exclusive access.

Being operating-system-independent is important for secure data. KeePassX runs on GNU/Linux, BSD, Mac and Windows. KeePassDroid will put both user name and password into the notification drawer for easy selection on the phone.

You need to have backups and data liberation. Since KeePass files are encrypted, copying them elsewhere is sufficient for backups. Keep old master passwords secure as they can be used on old backups of the file.

Data liberation means you can access your data even if the original application or service provider stops working. A standard format like KeePass works since many applications can read it. Additionally, KeePassX exports to a plain-text file. Make sure to protect it with an encryption tool, such as GnuPG.

For online password managers, data liberation means you can get your data back out of the service provider in an open format. If you use a proprietary service, you are dependent on that service allowing you access. Make regular exports to an open format in case the provider goes offline, decides you haven't paid recently or jumbles your data.

I recommend separate files for personal and work. No need for a subpoena about some work contract to have access to all of your personal credentials. Just like email, keep personal and work in separate accounts.

For traveling or mobile use, you don't need all of your credentials. You can drop the whole file on some file-sharing service and unlock the entire file from your phone, but then a compromise on your phone has access to all of your accounts as well.

Keep an updated file with a subset on the mobile device. I recommend the minimum essentials to connect to your file-sharing service and operate if you have voice, but not data.

KeePassX does not provide a way to sync data to another location securely, but `kpcli` can export a group to a new KeePass 1.x file. For instance, create a "mobile" group in your KeePass file, then use `kpcli` to export that group. `kpcli` is available under the same licensing as Perl.

### Other Useful Features

A credential manager also can be used to escrow data like family social security numbers, important private dates, insurance account information

and anything else your family might need should you no longer be available. KeePassX also can store attachments. Now you have a data escrow manager. Power up.

A credential manager should liberally allow text for the master password. KeepPass files do. As expected, KeePass accepts special characters. In fact, when I give presentations on KeePassX, I often demo unlocking a KeePass file with cut and paste of Perl and shell script snippets.

KeePassX has an Auto-Type feature that will log you in to a website with one hot key: Ctrl-v. Although other people like Auto-Type features, I find it too easy to mis-click and paste my credentials in the wrong place. Eventually I do get it right, usually long before “user+twyimCij5-fsf@example.com zFsZ5ZwEh5FHRgYf474MCRQ8pW4YNDXT87wrsQhkyL” starts trending in Google searches.

KeePass files also support an expiration date for passwords, and KeePassX provides preset time periods like three and six months. Once the entry expires, KeePassX marks it when viewing the group as a reminder to change the password.

KeePass allows you to access previous versions of an entry via its history feature. It's not quite revision control, but it's still handy.

It's also handy to provide a means of live security and authenticity testing.

For instance, LastPass is an online commercial password manager with a history of quickly responding to needs. For example, shortly after Heartbleed was announced, LastPass integrated a check to verify the destination website did not have the Heartbleed bug before sending credentials. It also does some verification to make sure you're connecting to the actual site rather than some phishing site.

LastPass is mostly browser-based. The company also provides a command-line tool with source available, and a perusal of several files yielded GPLv2 or later licensing.

LastPass does encryption and password creation on the client side, so the company never has access to your unencrypted data. It shares credentials by encrypting on the client side with the recipient's public key. Again, LastPass doesn't see unencrypted credentials.

LastPass also advertises a feature that allows the other person to use

the credentials without seeing them. Although LastPass can't view the unencrypted data, it's fairly easy for the recipient to capture the hidden plain-text password.

See the Password Managers sidebar for a list of free software password managers.

If you prefer password generation and password storage to come from different applications, the Generating Random Text sidebar has some ideas for creating your own random strings. ■

---

**der.hans** is a free software, technology and entrepreneurial veteran. His roles have included director of engineering, engineering manager, IS manager, community college instructor, developer, DBA and his favorite, system administrator. He is also now a repeat author for *Linux Journal*. In his free time, der.hans endeavors to help build the Free Software community through user group and conference leadership. He is chairman of the Phoenix Linux User Group (PLUG), BoF organizer for the Southern California Linux Expo (SCaLE) and founder of the Free Software Stammtisch. He's currently supprting manufacturing in the US as a senior engineer at Shutterfly.

### RESOURCES

KeePassX Licensing: <https://github.com/keepassx/keepassx/blob/master/COPYING>

KeePassDroid Licensing: <https://f-droid.org/repository/browse/?fdid=com.android.keepass>

Automated Password Generator (apg): <http://www.adel.nursat.kz/apg>

LastPass: <https://www.LastPass.com>

Sub-addressing RFC: <https://tools.ietf.org/html/rfc5233>

*LJ* article on KeePassX 1.x—“KeePassX: Keeping Your Passwords Safe” by Anthony Dean: <https://www.linuxjournal.com/content/keepassx-keeping-your-passwords-safe>

Mat Honan: <https://www.wired.com/2012/08/apple-amazon-mat-honan-hacking>

Send comments or feedback via  
<http://www.linuxjournal.com/contact>  
or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

[RETURN TO CONTENTS](#)



# GENERATING RANDOM TEXT

The biggest key to a strong password is entropy, randomness. Use long, pure random text to get high entropy. A credential manager isn't the only way to create long strings of random text. For instance, let your cat walk on your keyboard.

Here are some methods that are more efficient and less scratchy than trying to train your cat to work on demand:

1) Use a random text generation tool, such as `apg`, `pwgen` or `makepasswd`.

2) Use GnuPG's random function:

```
$ gpg --gen-random --armor 1 42
```

3) Use OpenSSL's random function:

```
$ openssl rand -base64 42
```

4) Use `uuidcdef` to create a UUID:

```
$ uuidcdef -u
```

5) Create a hash of a real answer plus a master password as the salt:

```
$ echo "anteater twiddle aquamarine drove \  
LinuxJournal.com" | sha256sum | cut -c 42
```

A feature of this method is that one master password can derive any site password without saving anything to disk. There's even a tool to help with that:

```
$ passwordmaker -r LinuxJournal.com -g 42 -a SHA1 \  
-x -m "anteater twiddle aquamarine drove"
```

One problem with derive-on-the-fly is that you don't have per-site random text. `passwordmaker` does allow using an unencrypted configuration file.

6) Use `date` to generate a random birthdate:

```
$ date -d @$((($RANDOM*24*3600/2-500000000)) +%Y%b%d
```

# EMAIL SUB-ADDRESSING

Sub-addressing augments your email address by inserting tokens. It turns one email address into as many as you want.

A separator character is placed after the user name portion of an email address, and then you can add a token, <username><character><token>@<domain>. You decide on the token.

Tokens can protect your accounts as listed in the article and also filter email. For instance, phishers like to send fake eBay mail. It's not eBay's fault, but it's a popular target. With an eBay token, you know that email without the token, user+pl8qr-ebay@example.com for instance, isn't from eBay and can be trashed automatically.

Some common email providers and their separator characters are:

- Gmail: +
- FastMail.FM: +
- Kolab: +
- Yahoo!: -

Gmail sub-addressing can be disabled at the domain level, so it might not be available for work accounts.

FastMail.FM also offers sub-domain as an alternative to sub-addressing.

# EMAIL SUB-ADDRESSING ON YOUR OWN SMTP SERVER

Some common SMTP servers and their default separator characters are:

- Postfix: +
- Exim: default off
- Courier and qmail: -

Some considerations if adding sub-addressing to a domain: many websites disallow a plus sign in an email address; periods are allowed everywhere, but often they already are in use for first.last@domain; dash is also fairly universally allowed.

## PASSWORD MANAGERS

Many free software password managers are available:

- KeePassX: <https://www.KEEPASSX.org>
- KeePassDroid: <http://www.KEEPASSDROID.com>
- kpcli: <http://kpcli.sourceforge.net>
- KeePass: <http://www.KEEPASS.info>
- Password Safe: <https://pwsafe.org>
- pass: <https://www.PASSWORDSTORE.org>
- Console Password Manager (cpm): <http://github.com/comotion/cpm>
- pwman3: <https://pwman3.github.io/pwman3>

KDE and GNOME have built-in password managers, and there are web browser plugins.

# LOW POWER WIRELESS: CoAP

This article concludes the series on 6LoWPAN by looking at the application layer: device description using CoRE Link Format, data formats using CBOR and REST APIs. Although these are only one set of choices, they are emerging as the principal choices for low power devices. I use Python to illustrate these, but many other languages are possible.

JAN NEWMARCH

---

PREVIOUS



Feature: Online Privacy  
and Security Using  
a Password Manager

NEXT  
Doc Searls' EOF



In my previous two articles (see the November and December 2016 issues of *LJ*), I described setting up a 6LoWPAN network and then integrating that into the internet so that the low power sensor/actuator nodes can talk to internet hosts and vice versa. This is one of the major mechanisms currently proposed for bringing the Internet of Things (IoT) to life.

Once you have established a communications pathway, however, you need to look at how you are going to use that pathway to exchange information—specifically, the protocols and the data types. The currently favored protocols are MQTT (MQ Telemetry Transport) and CoAP (Constrained Application Protocol), and they each fill different roles. MQTT is a messaging system using publish/subscribe, which has been adapted for low power devices. CoAP is similar to, and based on, HTTP but is heavily optimized for low power devices. This article focuses on CoAP.

### CoAP

The World Wide Web is built on the HTTP protocol. This is a traditional client/server model, where clients connect to a server over TCP and make requests of the server, which in turn prepares replies and delivers them to the client. The outstanding success of the Web has led to this being used as the model for CoAP, with the following appropriate changes:

- HTTP is TCP-based. A TCP session requires a handshake setup, acknowledgment of packets, retries on failure and keepalive mechanisms. UDP is much lighter; packets are sent via a “send and forget” mechanism. CoAP uses UDP, and it is up to the sender as to whether it requires an acknowledgement.
- Because HTTP is a transport protocol, any application-layer protocol can be built on top of it. One of the worst of these was SOAP, a protocol that has had some success in enterprise systems. CoAP uses REST, which is much closer aligned with HTTP.
- Application data traditionally has been attached to HTTP packets using formats like XML and, more recently, JSON. These are text-based,

and consequently, they are heavy both in payload and in processing. CoAP applications can use any of these, but the trend is toward CBOR (Concise Binary Object Representation), a binary version of JSON.

- HTTP does not allow messages to be sent from the server to clients, so CoAP has added a mechanism for this. This is coming into HTTP through “server push” mechanisms.

## REST

REST (REpresentational State Transfer) is the philosophy behind HTTP, described by Roy Fielding (the principal HTTP 1.1 architect) in his PhD thesis. In a horribly emasculated form, he says that 1) resources are identified by URIs, such as Web URLs, and 2) resources are accessed using only four verbs: GET, PUT, POST and DELETE, with defined meanings (although when to use PUT and when to use POST is still debated). These definitions are:

- GET: get a representation of a resource. For sensor data, this most likely will be in JSON or CBOR format, and will contain data, such as the temperature of a sensor.
- PUT: set a new value for a resource. For a heating system, it could be setting a new temperature value.
- POST: usually used to create a new resource, and possibly of limited use for sensors and actuators.
- DELETE: delete a resource. Again, it’s possibly of limited value here. I provide a more expansive version at <https://jan.newmarch.name/IoT/Middleware/REST>.

## CoAP and Python

CoAP will most likely be run as a server on sensors and actuators. These won’t be highly endowed with RAM, and they are actually unlikely to be able to run Python, which takes megabytes of RAM. Even micro-Python takes about 180kB of RAM. Most likely, they will run compiled code, using

a library such as the libcoap C library (<https://libcoap.net>).

I'm running these examples on Raspberry Pis, so I'm using Python for simplicity. There are many Python packages for CoAP and many implementations of CoAP for other languages (see <http://coap.technology/impls.html>). The Ubuntu x86 repositories have the aiocoap Python package, so you can install that on your desktop with:

```
sudo apt-get install python3-aiocoap
```

The RPi repositories currently have no CoAP packages, so you will have to install something (again!). You need to get the CoAP package on the sensor RPi. Download it from aiocoap, the Python CoAP library (<https://github.com/chrysn/aiocoap>). It contains the Python libraries in the aiocoap directory as Python code. You can move that directory to, say, /usr/lib/python3.4, so that it can be found from any Python 3 program:

```
git clone --depth=1 https://github.com/chrysn/aiocoap.git
cd aiocoap/
sudo mv aiocoap /usr/lib/python3.4
```

The package also contains clientGET.py, clientPUT.py and server.py. These not only demonstrate the CoAP package, but they also test some features. (I'll adapt these to our purpose here.)

## A Simple CoAP Application

I'm going to use the CPU temperature example from my previous two articles, as it is about as simple as one can get.

The sensor has to be exposed as a resource—that is, have a URI (here a URL). This will use the scheme `coap://` or `coaps://`, its IPv6 address and its URI path, such as `temperature`. Note that the sensor will be running as a *server*—the *client* will be making queries to the server.

To the client, the URL will look like this:

```
coap://[fd28::2]/temperature
```

using the global IPv6 address you set on the "sensor" RPi in the

previous article. The IPv6 address needs to be in square brackets ([...]) to avoid the colons (:) being confused with a URL Port option. The default UDP port is 5683.

The aiocoap package uses the recently added `yield` from Python generator system. I won't go into that here (it is non-trivial). The major parts to note are what you configure in the client and server.

The client needs to set the method as `GET` to fetch the CPU temperature of the server, using the server's URI. Then it reads a response and does something to it. Here you just print the response. The client is:

```
#!/usr/bin/env python3

import asyncio

from aiocoap import *

@asyncio.coroutine
def main():
    protocol = yield from Context.create_client_context()

    request = Message(code=GET)
    request.set_request_uri('coap://[fd28::2]/temperature')

    try:
        response = yield from protocol.request(request).response
    except Exception as e:
        print('Failed to fetch resource:')
        print(e)
    else:
        print('Result: %s\n%r'%(response.code,
                                response.payload.decode('utf-8'))))

if __name__ == "__main__":
    asyncio.get_event_loop().run_until_complete(main())
```

The server uses the asynchronous I/O package `asyncio`. Again, you can



ignore the details of this. The important thing is to add *resources* that can be accessed by CoAP user agents (clients). You add a new resource with:

```
root.add_resource(('temperature',), TemperatureResource())
```

which sets the URI-Path (/temperature) of the resource on this host and a class (`TemperatureResource`) to be invoked when the resource is requested. Any number of resources can be added, such as pressure, motion and so on, each with their own class handler.

The handling class is the most complex, and there are many possibilities. The simplest will subclass from `aiocoap.resource.Resource` and will have a method `render_get`, which is called when a GET for a representation of the resource is needed. For the example sensor, this gets the CPU temperature as before and then wraps it into an `aiocoap.Message`. Here's the server code:

```
#!/usr/bin/env python3

import asyncio
import aiocoap.resource as resource
import aiocoap

from subprocess import PIPE, Popen

class TemperatureResource(resource.Resource):
    def __init__(self):
        super(TemperatureResource, self).__init__()

    @asyncio.coroutine
    def render_get(self, request):
        process = Popen(['vcgencmd', 'measure_temp'], stdout=PIPE)
        output, _error = process.communicate()
        return aiocoap.Message(code=aiocoap.CONTENT, payload=output)

def main():
    # Resource tree creation
```

```

root = resource.Site()

root.add_resource(('temperature',), TemperatureResource())

asyncio.async(aiocoap.Context.create_server_context(root))

asyncio.get_event_loop().run_forever()

if __name__ == "__main__":
    main()

```

The output from running the client against this server is similar to this:

```

Result: 2.05 Content
"temp=36.9'C\n"

```

as in previous examples.

## Making Things Reusable

What I've basically done at this point is hack up an example to show how CoAP works, but the IoT isn't going to succeed if programmers act like that. My sensor will need to work in your environment, talking to other people's systems. The IoT isn't going to be a simple monolithic environment. It's going to be a mess of multiple systems trying to talk to each other.

Standards and conventions will need to be agreed upon, and not just between people, but in ways that can be read and confirmed by machines. I've used CoAP over 6LoWPAN, and that is just one battle that is raging. The next one is over data formats and device descriptions—both using them and discovering them.

## Data Formats

HTTP has mechanisms to query and to specify data formats. For HTTP, this is managed by Content Negotiation, and this idea is carried across into CoAP: a client can request particular data formats, while the server may have preferred and default formats.

XML is generally regarded as too heavyweight. JSON is better, but as a text format, it still carries baggage. CBOR (Concise Binary Object Representation) is an IETF RFC for a binary encoding of JSON and is becoming popular. It has an advantage of being self-contained, unlike other recent binary formats, such as Google's Protocol Buffers, which require an external specification of the data.

A JSON format of the sensor data could look like this:

```
{
  "temperature" : 37.9,
  "units" : 'C'
}
```

CBOR translates this into a binary format, which may be more concise.

To use CBOR, first you need to install it. Python packages normally are installed using `pip`, and the RPi does not come with this installed. So install both it and the `cbor` module (note that you want the Python 3 versions):

```
sudo apt-get install python3-pip
sudo pip3 install cbor
sudo pip3 install LinkHeader
```

Then, a JSON equivalent data type can be encoded using `cbor.dumps`, which creates a byte array and is decoded by `cbor.loads`, which turns it back into a Python type. A Python dictionary is equivalent to the JSON of a JavaScript class object given above.

The server is modified by code to create a Python dictionary and then turn it into CBOR. The client is likewise modified to decode the CBOR data into a Python dictionary. You also will do some elementary content specification, using IANA-registered numbers. The `application/cbor` number is 60, from the IETF RFC 7049.

The relevant part of the server is this:

```
CONTENT_FORMAT_CBOR = 60
```

```

class TemperatureResource(resource.Resource):
    def __init__(self):
        super(TemperatureResource, self).__init__()

    @asyncio.coroutine
    def render_get(self, request):
        process = Popen(['vcgencmd', 'measure_temp'], stdout=PIPE)
        output, _error = process.communicate()
        list = re.split("[=\n]", output.decode('utf-8'))
        dict = {'temperature' : float(list[1]), 'unit' : list[2]}
        mesg = aiocoap.Message(code=aiocoap.CONTENT,
                               payload=cbor.dumps(dict))
        mesg.opt.content_format = CONTENT_FORMAT_CBOR
        return mesg

```

And, here's the relevant part of the client:

```

request = Message(code=GET)
request.set_request_uri('coap://[fd28::2]/temperature')

try:
    response = yield from protocol.request(request).response
except Exception as e:
    print('Failed to fetch resource:')
    print(e)
else:
    if response.opt.content_format == CONTENT_FORMAT_CBOR:
        print('Result: %s\n%r'%(response.code,
                                cbor.loads(response.payload)))
    else:
        print('Unknown format')

```

This prints something like this:

```

Result: 2.05 Content
{'temperature': 37.4, 'unit': 'C'}

```

## Device Descriptions

The code above is fine for interacting with a temperature sensor—once you know what that is! You may have hundreds of sensors of different types, and all you may know is their IPv6 address. To complete this, you need to know the following:

- What is the specification of a device, such as a “temperature sensor”?
- What are the special values for your sensor (for example, max and min temperatures)?
- How do you tell what type of device you have?
- How do you know how the CoAP requests interact with your device?

At the moment, there are no industry-agreed-upon answers to those questions. One could say that (unfortunately) this is another of the differentiators in the IoT world. The IETF in RFC 7252 and RFC 6690 has made some progress, but there are still open issues, and they are not uniformly adopted.

From RFC 6690, each device should have a URI-path of `/.well-known/core`, which can be accessed by an HTTP `GET coap://<IPv6-addr>/.well-known/core` request. RFC 6690 specifies that the representation must be in CoRE Link Format, which I will describe soon.

Two new link attributes are added to the standard Web link headers of RFC 5988, such as `title`. The new attributes are the following:

- `rt` for resource type.
- `if` for interface type.

The values of these attributes can be strings, URLs or anything—this isn’t specified. The resource type is expected to be some “well known” value that identifies the type of device, such as `jan.newmarch:temperature-sensor`. Yes, I just made that up—there are several proposals but no standards yet.

The value of `if` is supposed to be some specification of the REST

interface for the device—that is, how to call it using **GET**, **PUT** and so on, and what is returned from those calls. How an interface is described isn't specified by RFC 6690. Although possibly using WADL (Web Application Description Language) is suggested, the Open Connectivity Foundation uses RAML (RESTful API Modeling Language), and the Wikipedia page on RESTful APIs lists a dozen more, probably used by some group or other.

Investigating REST API languages is beyond the scope of this article, so let's just assume the well known core resource has a value like this:

```
</temperature>;rt="jan.newmarch:temperature-sensor";
    if="https://jan.newmarch.name/temperature-sensor"
```

Here `/temperature` is the relative URL of the resource, the value of `rt` is the “well known” device type, and the value of `if` is the description of the device. Assume that `https://jan.newmarch.name/temperature-sensor` contains WADL or RAML or some other description that allows you to deduce that requesting the resource `/temperature` using **GET** will return a CBOR object with fields `temperature` and `unit`, with float and string values, respectively.

The format of the well-known resource is defined to be in `application/link-format`, which according to the IANA CoAP Content-Formats site (<https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#content-formats>) has CoAP code 40. The format is actually just UTF-8.

The server is modified by adding another resource:

```
root.add_resource(('.well-known', 'core'), WKResource(root))
```

where `WKResource` is a class in the `aiocoap` module, which keeps a list of all the resources supplied by this device.

When the client **GETs** the resource `/.well-known/core`, it will get a comma-separated list like this:

```
</.well-known/core>; ct=40,
</temperature>;
    if="https://jan.newmarch/temperature-sensor";
    rt="jan.newmarch.name:temperature-sensor"
```

For each resource, a client should extract the `rt` value. If it recognizes it as a temperature device, then it should carry on. If it doesn't, it should look up the `if` URL and extract what the `GET` method can do, and then carry on. That code is not covered here. The CoRE Link Format string can be parsed using the Python LinkHeader package.

## Conclusion

This series has addressed the issues of setting up a 6LoWPAN low power wireless network, using the OpenLabs radios on Raspberry Pis, followed by bringing these devices into internet visibility. This concluding article looks at data formats and protocols for the IoT.

Many topics have been omitted. The major one is that of security, as the system I have described here is wide open to snooping and hacking. The security mechanisms are all there, but they are a full topic in their own right.

I also have ignored the issue of how external clients find the IP addresses of the clients. This is answered by internet draft "CoRE Resource Directory draft-ietf-core-resource-directory" (<https://tools.ietf.org/html/draft-ietf-core-resource-directory-09>).

I haven't addressed networking within a 6LoWPAN network. There are a variety of models, such as mesh networking, and they build on the IEEE802.15.4 networking model.

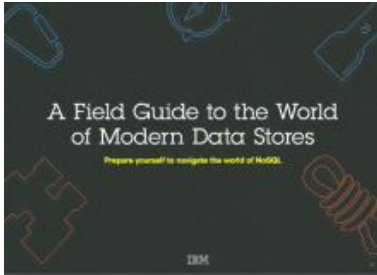
Finally, I haven't mentioned other pieces of hardware dealing with IEEE802.15.4 and 6LoWPAN. These include modules from Texas Instruments, Firefly and Libelium, with many others coming along. ■

---

**Jan Newmarch** has been using Linux since kernel 0.96. He has written many books and papers about software engineering, network programming, user interfaces and artificial intelligence, and he is currently digging into the IoT. He is in charge of ICT degrees at Box Hill Institute and Adjunct Professor at the University of Canberra.

Send comments or feedback via  
<http://www.linuxjournal.com/contact>  
or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

[RETURN TO CONTENTS](#)



## A Field Guide to the World of Modern Data Stores

There are many types of databases and data analysis tools to choose from when building your application. Should you use a relational database? How about a key-value store? Maybe a document database? Is a graph database the right fit? What about polyglot persistence and the need for advanced analytics?

If you feel a bit overwhelmed, don't worry. This guide lays out the various database options and analytic solutions available to meet your app's unique needs.

You'll see how data can move across databases and development languages, so you can work in your favorite environment without the friction and productivity loss of the past.

Sponsor: IBM

> <https://geekguide.linuxjournal.com/content/field-guide-world-modern-data-stores>

---



## Why NoSQL? Your database options in the new non-relational world

The continual increase in web, mobile and IoT applications, alongside emerging trends shifting online consumer behavior and new classes of data, is causing developers to reevaluate how their data is stored and managed. Today's applications require a database that is capable of providing a scalable, flexible solution to efficiently and safely manage the massive flow of data to and from a global user base.

Developers and IT alike are finding it difficult, and sometimes even impossible, to quickly incorporate all of this data into the relational model while dynamically scaling to maintain the performance levels users demand. This is causing many to look at NoSQL databases for the flexibility they offer, and is a big reason why the global NoSQL market is forecasted to nearly double and reach USD3.4 billion in 2020.

Sponsor: IBM

> <https://geekguide.linuxjournal.com/content/why-nosql-your-database-options-new-non-relational-world>

---



## RunKeeper Case Study

Boston-based fitness start-up RunKeeper was struggling with its database, and could not keep pace with the company's expansion. With new users joining every day, this limitation threatened to halt the company's operations. With a database of 30 million users and growing fast, scaling up also became an issue.

RunKeeper's initial database PostgreSQL failed to provide the required speed and scale. Partnering with IBM, RunKeeper transformed using IBM Cloudant's Dedicated Cluster as its new data layer.

"We were impressed by the wealth of experience that the IBM team was able to draw on to adapt the solution to meet our business needs," says Joe Bondi, CTO and Co-founder of RunKeeper.

Sponsor: IBM

> <https://geekguide.linuxjournal.com/content/run-keeper-case-study>





## The 2016 State of DBaaS Report: How managed services are transforming database administration

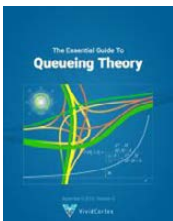
If you didn't have to manage your database, what would you do with your free time? All those hours you previously spent micromanaging your data layer—ensuring it keeps your application running 24/7 and is able to scale up or down based on demand— would suddenly reappear in your day. You could spend more time building your applications, from adding key features to improving the experience of your users, and you would even get some hours back in your personal life.

The 2016 State of DBaaS Report, commissioned by IBM, assessed the business and technical impact of database-as-a-service (DBaaS), as identified by 680 executive and technical enterprise users, and found that developers are saving a substantial amount of time after adopting DBaaS. All of those surveyed were using a managed, NoSQL database service across a variety of industries, including insurance, healthcare, gaming, retail and finance.

Sponsor: IBM

> <https://geekguide.linuxjournal.com/content/2016-state-dbaas-report-how-managed-services-are-transforming-database-administration>

---



## The Essential Guide To Queueing Theory

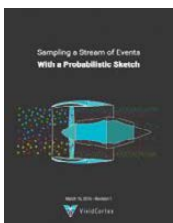
Whether you're an entrepreneur, engineer, or manager, learning about queueing theory is a great way to be more effective. Queueing theory is fundamental to getting good return on your efforts. That's because the results your systems and teams produce are heavily influenced by how much waiting takes place, and waiting is waste. Minimizing this waste is extremely important. It's one of the biggest levers you will find for improving the cost and performance of your teams and systems.

Author: Baron Schwartz

Sponsor: VividCortex

> <https://geekguide.linuxjournal.com/content/essential-guide-queueing-theory>

---



## Sampling a Stream of Events With a Probabilistic Sketch

Stream processing is a hot topic today. As modern Big Data processing systems have evolved, stream processing has become recognized as a first-class citizen in the toolbox. That's because when you take away the how of Big Data and look at the underlying goals and end results, deriving real-time insights from huge, high-velocity, high-variety streams of data is a fundamental, core use case. This explains the explosive popularity of systems such as Apache Kafka, Apache Spark, Apache Samza, Apache Storm, and Apache Apex—to name just a few!

Author: Baron Schwartz

Sponsor: VividCortex

> <https://geekguide.linuxjournal.com/content/sampling-stream-events-probabilistic-sketch>

# Debugging Democracy

Let's start by making news reality-based again.



DOC SEARLS

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

---

PREVIOUS



Feature:

Low Power Wireless: CoAP

---

**Y**ou had to be a crank to insist on being right. Being right was largely a matter of explanations. Intellectual man had become an explaining creature. Fathers to children, wives to husbands, lecturers to listeners, experts to laymen, colleagues to colleagues, doctors to patients, man to his own soul, explained. The roots of this, the causes of the other, the source of events, the history, the structure, the reasons why. For the most part, in one ear out the other. The soul wanted what it wanted. It had its own natural knowledge. It sat unhappily on superstructures of explanation, poor bird, not knowing which way to fly.—Saul Bellow, *Mr. Sammler's Planet*, 1969.

I began writing this column on November 9, 2016, on the balcony of a hotel in Istanbul, while a call to prayer echoed through the streets below. I took that as good advice, because a few hours

earlier my country elected an Internet troll, Donald Trump, as its president (<http://scripting.com/2016/07/28/dontFeedDjTrump.html>). Perhaps by now we're calling this day 11/9, in the mold of 9/11. I'm an optimistic guy, but color me pessimistic about where my country is now heading, led by a world-class narcissist.

And forgive me for obsessing not only about where this is going, but how we got here. Our country has been hacked, and that matters.

Disclosure: I'm a political independent, and not a fan of Hillary Clinton, though I thought she was the only sensible choice, given Trump's shortcomings, many of which should have disqualified him, flat out. But he won. Why?

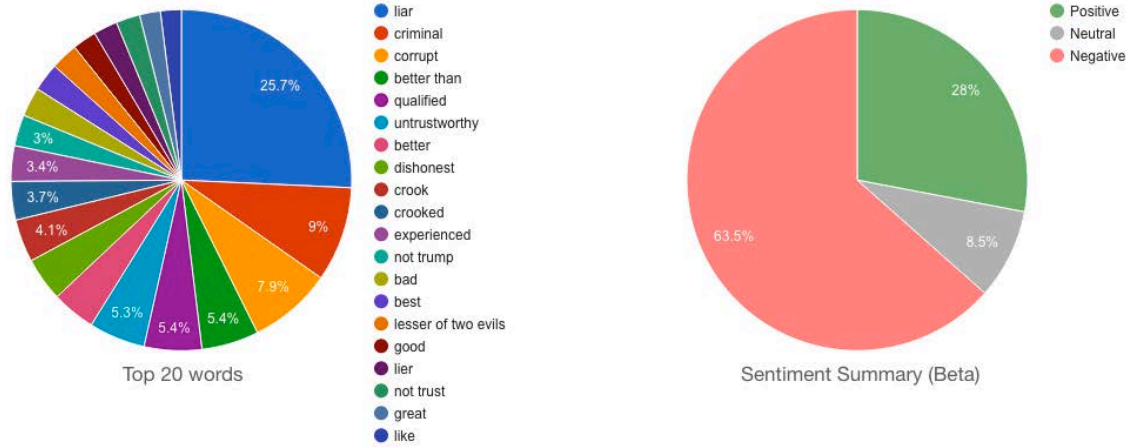
I don't know, though I did see it coming. Mostly I felt it. Polls said one thing, my senses another. "We know more than we can tell", says Michael Polanyi. Evidence: most of the time we don't know how we'll end the sentences we start, or how we started the sentences we end. Yet we know what we're talking about. And if we succeed, another human being gathers our meaning, even though they can't repeat it verbatim.

To say something is to express some care about it. We also tend to hear what we like to hear more than what we don't, even if we welcome what might disagree with us. Those of us who work with logic (such as *Linux Journal* readers) have a high regard for the rational. But while logic and reason sit on the mental board of directors, emotions cast the deciding votes. As Bellow says, the soul wants what it wants.

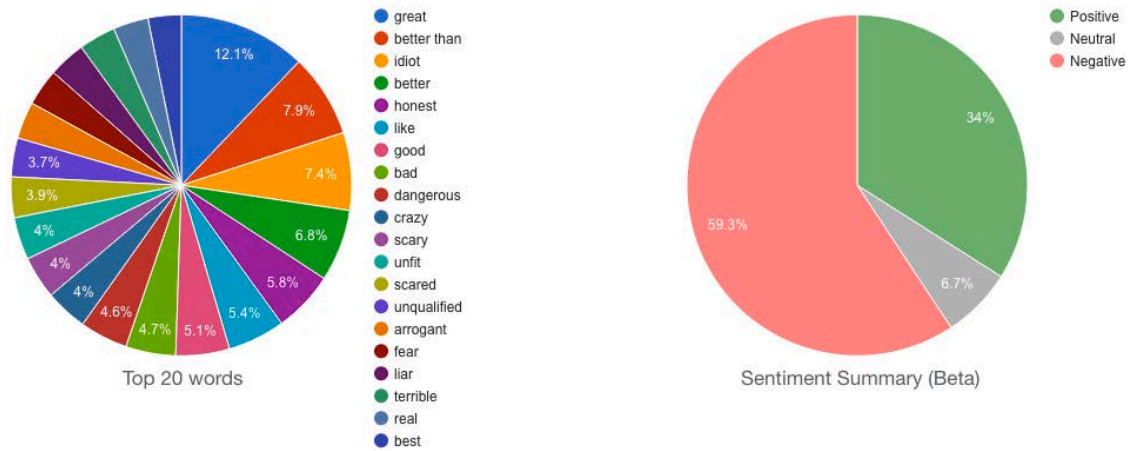
To see how emotions might cast deciding votes, Heartbeat AI studied emotional leanings in five "swing" states: ones a candidate needs to win in a close election (<https://heartbeatai.com/#welcome-1>). Before Election Day, ordinary polls showed Clinton winning most or all those states. But the Heartbeat AI study showed something very different. Here it is: <https://go.heartbeatai.com/report/client/529-241866ea30958b0b2007ffde94ddbbed9>. Play with it a bit. The little heart on the right is a tab that pulls out a drawer of variables you can turn on and off.

I just did that, and Figure 1 shows both candidates' sentiment

### Hillary Clinton



### Donald Trump



**Figure 1. Hillary Clinton and Donald Trumps’ Sentiment Maps for Five “Swing” States, via Heartbeat AI**

maps, together.

Trump won all those states. Hell, somebody had to. But the main point here is that voters disliked both candidates—a lot. They simply disliked Trump less than they disliked Clinton—not that they actually liked Trump.

This kind of study doesn’t show a mandate, but it does suggest caution before suggesting that a victory by either party constituted a mandate of any kind—or *should*, anyway.

It also makes one wonder how voters came to feel the way they did about the candidates. To what degree are those feelings attached to

actual facts? Hard to tell, but wondering should be productive.

On September 18, 2016, I blogged this (<http://doc.blog/2016/09/18/trumpTheMule.html>):

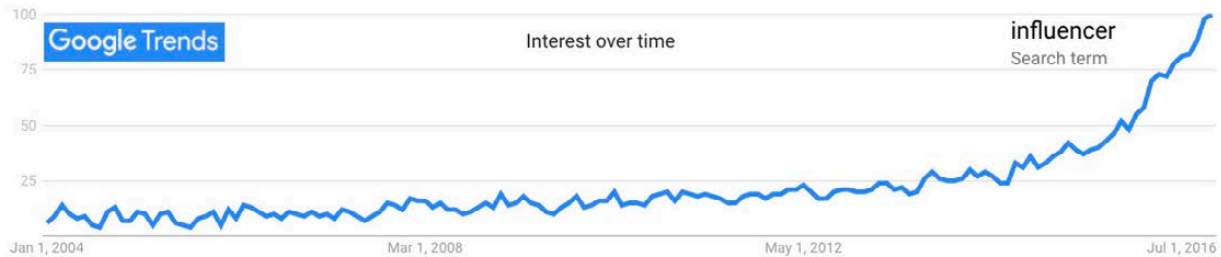
As soon as it became clear that Trump was a breed apart, remarkable more for his powers of persuasion and enlistment than for anything else (his policies are all feints: magical misdirections away from his absolute vanity), I saw him as the Mule ([https://en.wikipedia.org/wiki/Mule\\_%28Foundation%29](https://en.wikipedia.org/wiki/Mule_%28Foundation%29)), star of Isaac Azimov's *Foundation* series. Here's how Wikipedia describes the Mule:

"One of the greatest conquerors the galaxy has ever seen, he is a mentalic (<https://en.wikipedia.org/wiki/Mentalic>) who has the ability to reach into the minds of others and 'adjust' their emotions, individually or *en masse*, using this capability to conscript individuals to his cause. Not direct mind-control *per se*, it is a subtle influence of the subconscious; individuals under the Mule's influence behave otherwise normally—logic, memories, and personality intact."

Scott Adams more blandly calls Trump a "master persuader" (<http://blog.dilbert.com>). The effect is the same, especially if Trump wins. Which I fear. And, hate to say, expect.

Scott was right, and that surely makes Trump a much more interesting case study than Clinton, whose only quotable line during the whole campaign was "basket of deplorables". Her relatively blah public persona also made her a dressmaker's dummy for every adjectival characterization Trump chose to clothe her in, just as he had done with the most threatening of the 17 Republican opponents he knocked off in the primaries (examples: "low-energy Jeb", "little Marco", "Iyin' Ted"). Looking at the Heartbeat AI research, it seems the one that stuck in the final round was "Crooked Hillary".

The high degree to which Trump was disliked, even by people who voted for him, is nowhere in any of the prevailing narratives in election coverage, whether by (what's left of) mainstream media or by the millions yakking about it on Facebook and Twitter. Instead,



**Figure 2. Google Trends Results for “Influencer”**

we’re mostly hearing about how those deplorables won respect by defeating “the elites”.

Speaking of which, nearly all major newspapers sensibly opposed Trump, and most endorsed Clinton. Even some papers that normally favor Republican candidates either endorsed Clinton or suggested voting for anyone but Trump. In its 126-year history, the *Arizona Republic* never endorsed a single Democrat, but inveighed in favor of Hillary Clinton for this election (<http://www.azcentral.com/story/opinion/editorial/2016/09/27/hillary-clinton-endorsement/91198668>), calling Donald Trump a “two-bit billionaire” (<http://www.azcentral.com/story/opinion/editorial/2016/01/29/donald-trump-republican-imposteropinioneditorial/79417514>) and much worse.

Wikipedia’s list of Republicans opposed to Trump ([https://en.wikipedia.org/wiki/List\\_of\\_Republicans\\_who\\_opposed\\_Donald\\_Trump\\_presidential\\_campaign,\\_2016](https://en.wikipedia.org/wiki/List_of_Republicans_who_opposed_Donald_Trump_presidential_campaign,_2016)) and influencers in favor of Clinton ([https://en.wikipedia.org/wiki/List\\_of\\_Hillary\\_Clinton\\_presidential\\_campaign\\_endorsements,\\_2016](https://en.wikipedia.org/wiki/List_of_Hillary_Clinton_presidential_campaign_endorsements,_2016)) were both very long and thick with highly notable people and institutions. None of them mattered. Trump won anyway.

So much for the influence of influencers—or at least influencers of the usual kind. And it is interesting that this is all happening in a time when the term “influencer” is in marketing vogue (<https://www.google.com/trends/explore?date=all&q=influencer>). See Figure 2.

You’ll see almost exactly the same result for “influencer marketing” (<https://www.google.com/trends/explore?date=all&q=influencer%20marketing>), meaning that the main interest in “influencers” is among marketers. Not surprisingly, if you look up “influencer marketing”, you’ll get a load of results (nearly a half-million in my case, at that link), nearly all of it pitching shortcuts to earning influence. Says

**But I think there's more going on here. Before the internet came along, sources of authority were clear and understood.**

Wikipedia ([https://en.wikipedia.org/wiki/Influencer\\_marketing](https://en.wikipedia.org/wiki/Influencer_marketing)):

**Influencer marketing** (also **influence marketing**) is a form of marketing (<https://en.wikipedia.org/wiki/Marketing>) in which focus is placed on specific key individuals (or types of individual, <https://en.wikipedia.org/wiki/Individual>) rather than the target market as a whole. It identifies the individuals that have influence over potential buyers (<https://en.wikipedia.org/wiki/Buyer>), and orients marketing activities around these influencers.

Influencer content may be framed as testimonial advertising where they play the role of a potential buyer themselves, or they may be third parties. These third parties exist either in the supply chain (retailers, manufacturers, etc.) or may be so-called value-added influencers (such as journalists, academics, industry analysts, professional advisers, and so on).

As a journalist, I'm often targeted for manipulation by influence marketers. While I ignore or reject those appeals, I am also struck by the ironic sense that my own influence has been going down over recent years, roughly on a path that seems to mirror the upward curve in the graph shown in Figure 2. I am also sure the same is true for most journalists, if not all of them. Celebrities too. Endorsement failings in this election are just one example of that decline at work.

But I think there's more going on here. Before the internet came along, sources of authority were clear and understood. We had scientists, researchers, think tanks and other confirming or denying sources for claims, findings and stories told by politicians, business folk and other suspects given to lying and exaggeration. Good

journals and journalists could be trusted to find and expose essential facts, regardless of who got offended. Many fewer journalists are employed now, and many surviving journals have either gone out of business or morphed into the “content production” business. Most commercial publications (ours included) now show ads that are annoying, based on unwelcome tracking, or both, which reduces readership while eroding trust. Meanwhile, the old “mainstream media” are now just waves atop an ocean of “content” generated by anything and anybody.

Many journalists took great heart in the promise of the net and the web, back in the early years of both—especially how it gave everybody their own platform for publication and participation. In *We the Media* (<https://www.amazon.com/We-Media-Grassroots-Journalism-People/dp/0596102275>), first published in 2004, Dan Gillmor (<https://dangillmor.com>) details the progress of “journalism as lecture to journalism as conversation or seminar”. For example, take a post I put up on October 23, 2003, titled “Listen and Learn” (<http://doc.weblogs.com/2003/10/22#listenAndLearn>). There I visited some of my own learning experiences with journalism’s evolution in real time. Here’s an excerpt:

I find myself thinking there are three approaches to journalism represented here. One is the “cool” approach of traditional journalism, including network broadcasting (in which NPR is no exception). One is the “hot” approach of talk radio, which has since expanded to TV sports networks and now Fox TV. The third is the *engaged* approach of weblogging. What we’re doing here may be partisan in many cases, but it is also inconclusive. Blogging is about making and changing minds. It’s less about scoring points against perceived enemies—with certain exception (<http://www.rageboy.com/blogger.html>), of course—than about scaffolding new and better understandings of one subject or another.

Jay Rosen ([https://twitter.com/jayrosen\\_nyu](https://twitter.com/jayrosen_nyu)), a journalism professor at NYU, turned a one-liner from that passage into the title for a follow-up post: “Blogging is About Making and Changing Minds” ([http://archive.pressthink.org/2003/10/23/doc\\_inconclusive.html](http://archive.pressthink.org/2003/10/23/doc_inconclusive.html)). He



concludes that post with this:

The cool, neutral, professional style in journalism says: get both sides and decide for yourself. The hotter, more partisan press says: decide for yourself—which side?—then go get information. The weblog doesn't want to be either of these, but it checks and it balances both.

Those days are so gone.

So let's return to the noise—by which I mean Facebook.

At issue as I write this is “fake news” on Facebook, which seems to have influenced a lot of people in the election—or so says Craig Silverman in BuzzFeed ([https://www.buzzfeed.com/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook?utm\\_term=.mijmqmeLNG#.qylPQP34d1](https://www.buzzfeed.com/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook?utm_term=.mijmqmeLNG#.qylPQP34d1)),

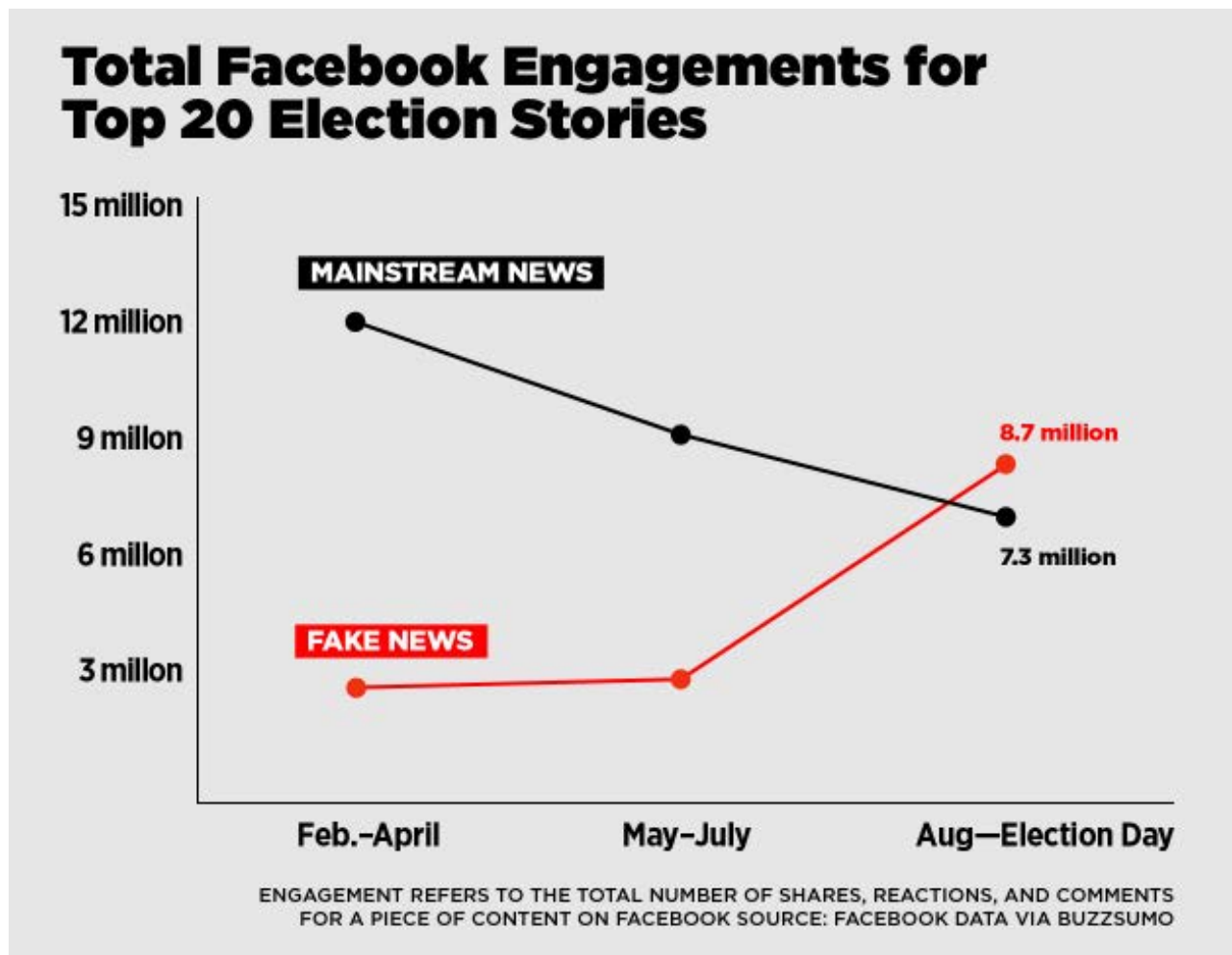


Figure 3. Influence of “Fake News” on Facebook (from Craig Silverman)

king of what we might call the “newstream” media. Craig did a bunch of research, most dramatically represented by the graphic shown in Figure 3.

I could go on, but I’d rather not, since by now I’ve spent more time on this piece than on anything I’ve written here in many years. (And I’ve been doing that since 1996.)

Here’s the Linux connection: **we need to hack news back in a logical direction, and away from the fact-free, misleading and emotion-stirring ways that news is made today.** The mainstream media is beyond fixing. So is the newstream media, so long as it remains dependent on surveillance-based advertising, clickbait and fake news of its own.

I don’t know how we do that, but we’ve hacked the world before: with free software, Linux and open source, just to name the Big Three. Time to do it again. ■

Send comments or feedback via

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

or to [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

[RETURN TO CONTENTS](#)

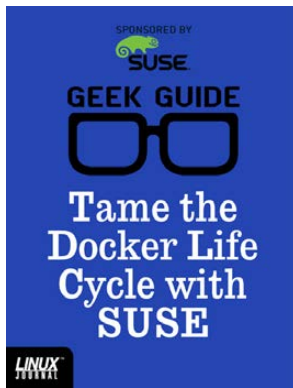
## ADVERTISER INDEX

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

ADVERTISER	URL	PAGE #
Drupalize.me	<a href="http://drupalize.me">http://drupalize.me</a>	116
Peer 1 Hosting	<a href="http://go.peer1.com/linux">http://go.peer1.com/linux</a>	21
SCALE 15x	<a href="http://www.socallinuxexpo.org">http://www.socallinuxexpo.org</a>	35
Silicon Mechanics	<a href="http://www.siliconmechanics.com">http://www.siliconmechanics.com</a>	15
SPTechCon	<a href="http://www.sptechcon.com">http://www.sptechcon.com</a>	19
SUSE	<a href="http://suse.com/storage">http://suse.com/storage</a>	7

### ATTENTION ADVERTISERS

The *Linux Journal* brand’s following has grown to a monthly readership nearly one million strong. Encompassing the magazine, Web site, newsletters and much more, *Linux Journal* offers the ideal content environment to help you reach your marketing objectives. For more information, please visit <http://www.linuxjournal.com/advertising>



## GEEK GUIDE

# Tame the Docker Life Cycle with SUSE

By John S. Tonello

It's no accident or mere passing fad that containers are revolutionizing how IT shops of all sizes do their work. Whether you're looking to make better use of existing data-center resources or improve portability to the cloud, Docker and the new-found freedom it offers to use virtual environments for everything from development to enterprise applications holds a lot of promise.

The challenge is figuring out how best to move beyond a standard Docker install to an enterprise-worthy solution that's secure, easy to manage and scalable. It's also important to find ways to manage all your containers easily as well as the images you modify and plan to reuse. After all, containers are only part of any enterprise, which is now a healthy mix of bare-metal boxes, virtual machines, containers and on- and off-premises clouds. Tools that can help provide a common framework—and familiar interfaces—are critical.

With SUSE Enterprise Linux Server 12 and the tools it offers, you and your team can begin to solve real-world problems, tame the Docker life cycle, and create, run and maintain containers at nearly any scale.

## The Container Revolution

Anyone managing hardware—from a

few blades to full data centers—knows that bare-metal server deployments are costly, time-consuming and not very efficient. Even if you could still afford it, the idea of running one or two services on a single physical server—maybe a database here, a website there—is just not practical. Even if you're the best system administrator out there, you can really make only educated guesses about the maximum amount of CPU, memory and storage a particular service will need over time. Once you do the math and purchase the hardware, you know there surely will be hours, days and weeks when your physical server's capacity is idle and of no use to you.

Virtual machines changed all that by enabling more efficient use of that same physical server's resources by sharing them across separate instances of Linux and Windows servers. With the advent of VMware and Hyper-V and open-source KVM and Xen, suddenly you could place multiple servers on a single physical box, quickly move them between clusters, more easily run backups and restores, clone them and manage them all from a single interface.

To continue reading, download the complete eBook for FREE at <http://geekguide.linuxjournal.com>.

# drupalize.me

## Instant Access to Premium Online Drupal Training

- ✓ *Instant access to hundreds of hours of Drupal training with new videos added every week!*
- ✓ *Learn from industry experts with real world experience building high profile sites*
- ✓ *Learn on the go wherever you are with apps for iOS, Android & Roku*
- ✓ *We also offer group accounts. Give your whole team access at a discounted rate!*

**Learn about our latest video releases and offers first by following us on Facebook and Twitter (@drupalizeme)!**

Go to <http://drupalize.me> and get Drupalized today!

