

# MUUGLines

The Manitoba UNIX User Group Newsletter

Volume 28 No. 5, January 2016

Editor: Trevor Cordes

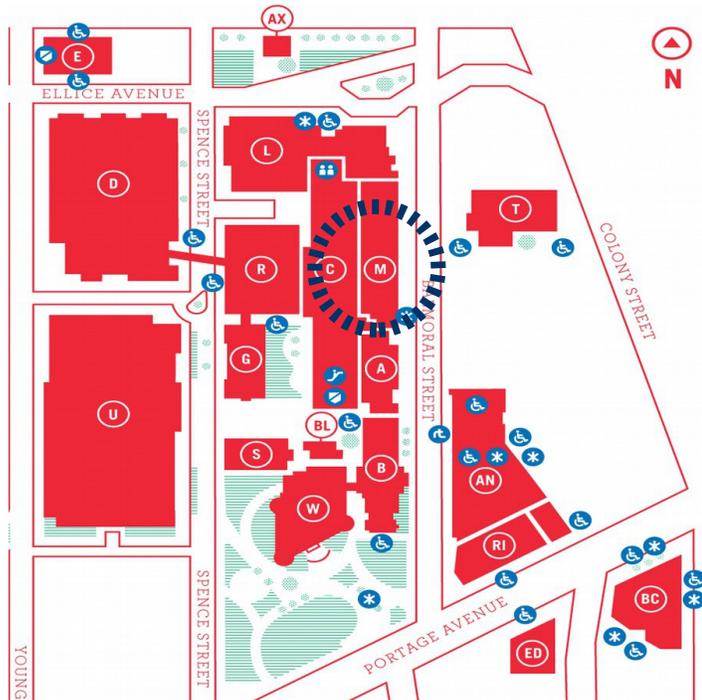
## Next Meeting: January 12<sup>th</sup>, 2016, 7:30 pm

### Main Topic

This month, Paul Sierks will be giving us an in-depth look at Logstash, which was briefly mentioned and demoed at the November meeting, when Rob Keizer presented it in the context of Elasticsearch and Kibana.

### RTFM

In the RTFM segment this month, Wyatt Zacharias will be showing us gencode(1).



## Where to Find the Meeting

### New Room! 3M60 Manitoba Hall

**New room!** This month's meeting is on the east side of campus in room 3M60, 3<sup>rd</sup> floor, Manitoba Hall, close to Balmoral Street. Look for a sign on the

door. There are elevators and escalators scattered around the buildings. A convenient one might be the elevator located right at the Ellice Ave entrance. Doors are usually open by 7:00 pm with the meeting starting at 7:30 pm. Parking is available on the surrounding streets. Please see <http://www.uwin-nipeg.ca/maps> for further information about parking and access to the campus.

**The latest meeting details are always at:**

<https://www.muug.mb.ca/meetings/>

## Infestation Insight

Get a handle on the quantity of bugs in your programming project. A new article on leankit demonstrates a mathematical way to estimate the number of remaining bugs based on the quantity found so far by different developers. It uses a tool called the "Lincoln Index".

Briefly: If A is the count of bugs found by tester A, and B the count for tester B (and there is no overlap); and C is the count found by both, then the estimate of the number of bugs remaining is given by  $AB/C$ . So for  $A=10$ ,  $B=20$ ,  $C=5$ , you can estimate 40 bugs remain. Interesting, and handy!

<http://tinyurl.com/jnlosc6>

## Firefox Flummoxes Freedom

Those upgrading to the latest Firefox (43.0) may notice a little glitch: some extensions may not function. Digging around you'll see that they are disabled because they are "unsigned". You'll only notice this if you use extensions downloaded from elsewhere other than the official Firefox extensions page; use (modified) extensions installed by your distro's package manager; use older/unsupported extensions; or use extensions you have written or modified yourself.

Sound familiar? This is akin to the Apple iOS model, where thou shalt only use Apple-approved downloads. Needless to say, this has riled up many FLOSS advocates. It equates to a back door attempt to impose a DRM scheme on users. Even if the stated purpose is benign (protect users from malwar-esque extensions) the end result is a loss of freedom.

In Firefox 43 you may disable signing to regain the old functionality by going into about:config and finding xpinstall.signatures.required and toggling its value to false. However, Firefox devs have stated that option will disappear in Firefox 44, leaving a scant few months for everyone to figure out a solution to this.

Fedora maintainers are some of the first on the scene, since Fedora includes in its standard repos many Fedora-ized RPM versions of popular plugins. One potential option put forth is the ditching of “branded” Firefox for a slightly modified version. That could be confusing for users, to say the least.

This author wonders why Firefox can’t simply have an option for unsigned packages where the user is presented with a stern warning and has to jump through some hoops – much like browsers do with self-signed SSL certs – but can still get the unsigned extension to function.

Chime in if you care about this issue, because once Firefox 44 arrives, we may be stuck with it. If not, *all your base belong to us.*

<http://tinyurl.com/oeztm4a>

<http://tinyurl.com/jnqx2q5>

<http://tinyurl.com/hzg6rkt>

## Videos Of The Month

“Become a better programmer in 30 minutes”. In these three compelling videos Tom Scott outlines why not to use floating point types for money, why to always store datetimes in UTC, and why to always use UTF8. These are light videos laid out in layman’s terms that any computer aficionado can appreciate. If you watch only one, make sure it’s the UTC video (then ponder the insanity).

<http://tinyurl.com/ndexzv2>

## Recursion by Stealth

If the names Dijkstra, Naur and Van Wijngaarden mean anything to you, and you want to read an intriguing story of how recursion barely made it into the first computer-language-by-committee in 1960, read the article linked below (long).

A large portion of the Algol-60 committee was against recursion. But conspirators in the “Amsterdam plot” snuck a single sentence into the final report: “Any other occurrence of the procedure identifier denotes reactivation of the procedure.” A fascinating read...

<http://tinyurl.com/qereenw>

## COBOL in JavaScript

“Have you ever wanted to just cut and paste some of that legacy COBOL code from mainframe applications into your latest Web application? No?” Well, someone has created a COBOL bridge for Node.js. OMG... or ewww?

<http://tinyurl.com/o2nny35>

## Crazy Cheap VPS?

A company called Scaleway has started selling VPS-like servers for only 2.99EUR a month. These aren’t even VPS (shared servers), these are dedicated computers that you alone will occupy. They have custom designed and built their own servers that have 4 ARM cores, 2GB RAM, 50GB SSD. Apparently they are so popular they are temporarily sold out. (This is not an endorsement.)

<http://scaleway.com>

## Visual Internet Anonymity

So it’s not just your MUUGLines editor! A man named Jonathan Hirshon has managed to keep his photo off the internet despite being in the tech industry, and hardly reclusive. In fact, he says “I live on Facebook”.

He is now on a mission to foil the recent advances in automated facial identification and other technologies by Google and Facebook. He wants you to tag random images with his name. That should keep the algos confused.

Jonathan is so committed to his cause that when he met with a guy wearing Google Glasses he brought a “device to foil” them: a balaclava. Why does Jonathan care? “I have worked for a number of companies in the (information and security) space. You can leave it at that, and let people draw their own conclusions.”

Let all of us who still care about our privacy take heart that we’re not alone; and to those who think it is impossible, take note that it is not! As someone who has programmed Facebook and mobile apps, I can attest to the astonishing amounts of personal information companies are pilfering from (mostly) ignorant users.

<http://tinyurl.com/qf5dqvh>

## Antitrust(ish) Amazon Antics

Amazon acting like Microsoft used to? Nah, can’t be. O’Reilly Media, everyone’s favourite computer book publisher, informed us in late 2015 that Amazon was discontinuing their “send to Kindle” feature. It had allowed shoppers on the O’Reilly web site to send an ebook directly to their Kindle with one click. You can still load them on your Kindle, but you now have to jump through many more hoops.

One source has claimed “This goes into Amazon’s usual hassle-competitors category... This confirms my hunch that those \$50 tablets are being sold below cost to increase the consumption of Amazon stuff, so material from other sources must be dried up, perhaps slowly to avoid bad press.”

Makes sense, since Amazon also sells the ebook versions of O’Reilly books on its own website.

<http://tinyurl.com/zcsqtkw>

<http://tinyurl.com/go5bedz>

## perl of Wisdom

Reading through my spanking new Modern Perl 4<sup>th</sup> Edition dead tree (review forthcoming), even this old hand is learning new tricks:

Accept named parameters with defaults in a subroutine:

```
sub JoinMUUG {
    my %parms=(
        when=>'this month',
        payvia=>'paypal',
        @_);
    ...
}
&JoinMUUG(payvia=>'cash',mood=>'happy');
```

This works because a hash can be built by supplying pairs of scalars (a key and a value); and repeated keys (i.e. in @\_) overwrite earlier definitions. It also relies on the fact that perl flattens lists in lists (of any depth) to just a list, so that: @foo=(2,4,6) is equivalent to @foo=(2,(4,(6))).

To boot, key=>value syntax is just gravy: you can use simply key,value,key2,value2... It turns out => syntax does give you the added bonus of implying single-quoting of the key, assuming your key consists of only normal identifier characters.

Another useful feature in more modern versions of perl is the short-circuiting defined-or assignment operator: //=

```
my %parms=@_;
$parms{geekfactor}//= 'high';
```

So if no key geekfactor exists in the hash, one is created and is given the value “high”. Even better, since it’s short-circuiting, the right-hand side can be an expression (with function calls, etc), and it will only be evaluated if the left-hand side is undefined. This gives you another great way to institute defaults in your code, and it’s great for building a hash of cached values of complicated computations. Beautiful!

## cdparanoia Not Paranoid Enough

by Trevor Cordes

Faced with a dead DVD-RW in my (Windows, sorry) PC and the prospect of reconfiguring EAC (Exact Audio Copy CD ripper) to work with a new model, I decided to spend a hefty chunk of time perusing the Linux alternatives. I'd love to ditch using the Windows box for ripping, and love to be able to whip up time-saving scripts on Linux instead of GUiing.

Of course, when you think CD ripping on Linux, you think cdparanoia. Interestingly, after more than a decade it appears to still be the only game in town for the serious ripper: someone who counts their CDs in the thousands, not dozens or hundreds. But can cdparanoia function as well as EAC?

My methodology was to configure both programs to give similar results, rip a bunch of CDs in various states of disrepair, and use cmp to compare the results. I had numerous old rips I could compare with: ones I knew were perfect having used my decades-perfected EAC setup.

cdparanoia seemed promising at first, but as soon as I started hitting CDs with a few scratches (not even bad ones), it became clear that cdparanoia was not paranoid enough. It was plunking in (usually slightly, but sometimes wildly) different values when it found errors. All variations of the relevant options were tried, to no avail.

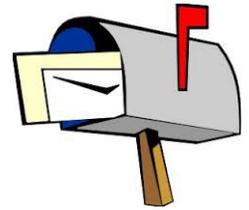
It did seem that cdparanoia was going to be ok when no errors showed up on its error-bar output line, but even that rule of thumb broke down eventually as I found a few CDs that would rip without error but still differ from EAC.

For testing cdparanoia I used both a modern SATA DVD-RW drive and an older, quality SCSI CD-RW. You could tell cdparanoia wasn't going to cut it partially by how quickly it would complete its rip compared to EAC.

Unfortunately, until Linux gets something like EAC released for it, this is one of those rare instances where closed-source Windows freeware outperforms Linux. Unless your CDs are pristine, you'll want to rip using EAC (or similar), after ensuring your setting configuration is spot on.

## MUUG Has A New Mailing Address

After more than 27 years MUUG has finally decided that renting a PO BOX for our physical mail is no longer justified (sorry Canada Post). Our new mailing address is now:



Manitoba UNIX User Group  
c/o Gilbert Detillieux  
Computer Science, E2-445 EITC  
University of Manitoba  
Winnipeg MB R3T 2N2

## MUUG has gone social!

For better or worse, you can now find (and follow!) MUUG on some social platforms. Get interactive and share all about MUUG on social media.



**Twitter:**

[twitter.com/manitobaunix](https://twitter.com/manitobaunix)



**Facebook:**

[facebook.com/ManitobaUnix](https://facebook.com/ManitobaUnix)



**MeetUp:**

[meetup.com/Manitoba-UNIX-User-Group](https://meetup.com/Manitoba-UNIX-User-Group)

## Creative Commons License



Except where otherwise noted, all content in this newsletter is licensed under a Creative Commons "Attribution-ShareAlike 2.5 Canada" License.

[http://creativecommons.org/licenses/by-sa/2.5/ca/deed.en\\_CA](http://creativecommons.org/licenses/by-sa/2.5/ca/deed.en_CA)