



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #70 - February 2013



Photo: mollybob (Flickr.com)



INSTALLING CODECS AND RIPPING DVDS IN UBUNTU

Full Circle Magazine is neither affiliated with, nor endorsed by, Canonical Ltd.



HowTo



Programming Python 40 p.08



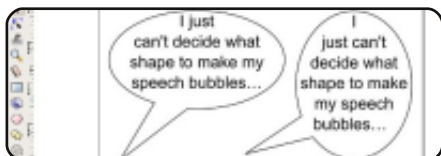
LibreOffice - Part 22 p.14

```
jeremy@HECTOR:~$ crontab -l
# Edit this file to introduce
#
# Each task to run has to be
# indicating with different
```

Cron p.17



Blender - Part 3 p.20



Inkscape - Part 10 p.23



Graphics



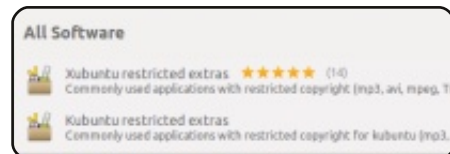
Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

Columns



Command & Conquer p.06



Ask The New Guy p.33



Linux Labs p.36



Linux Certified p.XX



Ubuntu News p.04



Ubuntu Games p.48



Q&A p.46



Ubuntu Women p.XX

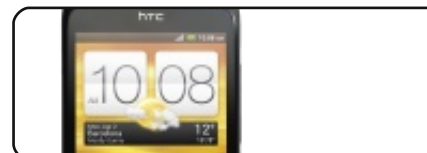
Opinions



My Story p.38



MyOpinion p.XX



Review p.41



Letters p.44



Web Dev p.27



@Web Dev



The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: you must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('Full Circle Magazine') and the URL www.fullcirclemagazine.org (but not attribute the article(s) in any way that suggests that they endorse you or your use of the work). If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

Full Circle magazine is entirely independent of Canonical, the sponsor of the Ubuntu projects, and the views and opinions in the magazine should in no way be assumed to have Canonical endorsement.



WELCOME TO ANOTHER ISSUE OF FULL CIRCLE!

This month, we have the usual Python, LibreOffice, Inkscape and Blender HowTo's. Joining them is an interesting HowTo on cron jobs. In other words: a script to do something on a particular day or time. In this month's Linux Labs, Charles looks at the strange world of DVD drives. Do manufacturers really hold back drive speeds? And for this month's review, we have Lucas telling us all about his new HTC One S phone.

I know I'm beginning to sound like a broken record (if you don't know what a 'record' is, ask your parents), but I need new articles. I'm left with very few desktop screens, no opinion articles, only a couple of 'my story' articles, and I have one review left from Art still to publish. So, now's the time to submit that article you've always wanted (or promised) to write. Have a read through our guidelines (<http://url.fullcirclemagazine.org/75d471>), then check the last page of this (or any) PDF issue for where to send your article. No articles means no magazine folks. No excuses. Everyone has something (hardware/software) that they can review. Everyone has a story of how they found Linux. Everyone likes to show off their pretty desktop (with info please!). And everyone has an opinion – keep it short and I can feature it on the currently ever shrinking Letters page.

All the best, and keep in touch!

Ronnie

ronnie@fullcirclemagazine.org



This magazine was created using :



Full Circle Podcast

Released monthly, each episode covers all the latest Ubuntu news, opinions, reviews, interviews and listener feedback. The Side-Pod is a new addition, it's an extra (irregular) short-form podcast which is intended to be a branch of the main podcast. It's somewhere to put all the general technology and non-Ubuntu stuff that doesn't fit in the main podcast.

Hosts:

- Les Pounder
- Tony Hughes
- Jon Chamberlain
- Oliver Clark



<http://fullcirclemagazine.org>



AUDIO MP3



AUDIO OGG

Download

THE UBUNTU FAMILY CONTINUES TO GROW; TABLETS NOW INCLUDED.

The Ubuntu website notes, “the Ubuntu family of interfaces now scales across all screens, so a phone can provide tablet, PC and TV experiences when docked.” The site also describes the Ubuntu tablet as a secure thin client that can be managed with the same

tools as any Ubuntu server or desktop which means that Ubuntu can now boast of a single OS for a phone, tablet, PC and TV.

<http://www.ubuntu.com/2013/02/press-pack-ubuntu-tablet>

UBUNTU DEVELOPER WEEK IS BACK

Daniel Holbach announces Ubuntu Developer Week, which will be held from 29th to 31st of January 2013. Holbach writes that, in the upcoming Ubuntu Developer Week, attendees can expect “sessions from hands-on introduction, packaging and Ubuntu development to talks about how to quickly get involved in certain teams and interact with other projects and much much more.”

<http://daniel.holba.ch/blog/2013/01/ubuntu-developer-week-is-back-2/>

SMART SCOPES

Jono Bacon announces increased development effort into the Unity dash and “Smart Scopes” which “will result in a significantly greater number of scopes (potentially 100) shipped by default and a better search experience that is smarter in the way scopes are searched for terms, thus delivering better results and higher performance.” Bacon also

notes that the Canonical Community Team “will also be launching a project over the coming few weeks to grow the range of scopes ready for 13.04 and ease the development process.”

<http://fridge.ubuntu.com/2013/01/30/smart-scopes/>

ANNOUNCING UBUNTU USER DAYS FEB 9-10TH

José Antonio Rey announces the Ubuntu User Days, which will be held February 9th and 10th, 2013 on IRC in [#ubuntu-classroom](#) (and [#ubuntu-classroom-chat](#) for questions and discussion). The schedule is available at <https://wiki.ubuntu.com/UserDays>

Everybody is welcome to participate and get their chance to learn about Ubuntu from Ubuntu community members.

<http://ubuntuclassroom.wordpress.com/2013/02/02/ubuntu-user-days-next-weekend/>



STEAM CLIENT NOW AVAILABLE IN UBUNTU SOFTWARE CENTER

The Steam client is now available in the Ubuntu Software Center, after Valve released Steam for Linux officially on Thursday, February 14th. To celebrate, all Linux titles in the Steam store are on sale with up to 75% off until February 21st. Users running Team Fortress 2 in Linux will also receive an in-game Tux penguin item, when they play, through the end of the month.



<http://blog.canonical.com/?p=2171>

UBUNTU 12.04.2 LTS RELEASED

Ubuntu 12.04.2 has been released, bringing with it many improvements, including an updated kernel and x stack, as well as the ability to install on UEFI systems with Secure Boot enabled. It also includes the updates since the last release, so that fewer updates will need to be downloaded after installation. In addition to Ubuntu 12.04.2 LTS,

Kubuntu 12.04.2 LTS, Edubuntu 12.04.2 LTS, Xubuntu 12.04.2 LTS, Mythbuntu 12.04.2 LTS, and Ubuntu Studio 12.04.2 LTS are now available as well.

<https://lists.ubuntu.com/archives/ubuntu-announce/2013-February/000166.html>

13.04 (RARING RINGTAIL) ALPHA 2 RELEASED

The second alpha of 13.04 Raring Ringtail has been released, which includes Kubuntu and Ubuntu Cloud images. The main Ubuntu version will not have alpha releases, however, opting instead for daily quality and nightly testing. The first milestone for Ubuntu will be the Final Beta Release, scheduled for the end of March. These pre-release versions should not be used by anyone needing a stable system, but should be free of "showstopper CD build or installer bugs."

<https://lists.ubuntu.com/archives/ubuntu-devel-announce/2013-February/001015.html>

Many Thanks to the Ubuntu News Team for their contribution this

month.

News this month comes from:

<https://wiki.ubuntu.com/UbuntuWeeklyNewsletter/Issue301>

<https://wiki.ubuntu.com/UbuntuWeeklyNewsletter/Issue302>

<https://wiki.ubuntu.com/UbuntuWeeklyNewsletter/Issue303>

<https://wiki.ubuntu.com/UbuntuWeeklyNewsletter/Issue304>





COMMAND & CONQUER

Written by Lucas Westermann

Suggestions

I haven't had many issues to fix this month, nor have I had any emails on topics I should cover. As such, I figured it could be time for another series of useful suggestions. This time around, I'll cover: recording electric guitar easily, using an android phone as a barcode scanner, transferring pictures from your android phone to your PC wirelessly, and a small tip for cable management.

RECORD ELECTRIC GUITAR

As many of you have probably heard, Ubisoft recently released a game called *Rocksmith*, which features a 1/4" jack to USB cable. Surprisingly enough, the cable seems to work without drivers on any OS (I've tested this on Mac OS X, Windows 8, and ArchLinux). So, for anyone who owns *Rocksmith* and an electric guitar – all you need to do is install Audacity and record! Due to the fact that the cable contains a normal 1/4" jack, you can run your signal through the usual chain (pedals, preamps, etc) before heading into your PC.



It's also possible to buy just the Real Tone Cable, though, for the price, you could probably find other solutions. If anyone is interested in a review of *Rocksmith* (for Xbox), send me an email and I'll get right on it.

Note: The quality of the recording is decent, but I found that it is sometimes too quiet. However, it's easily fixed with a bit of post-processing. Logically, the quality will also be dependent on the pickups in your guitar. All testing was done with my PRS SE Custom 24.

Sound samples (I apologize for the quality of the playing...):

<https://soundcloud.com/lswest/chords-pure> – Chords, played directly from the guitar into the PC
<https://soundcloud.com/lswest/santana-g3> – A little bit of a Santana tune, played through my Zoom G3 multi-effects pedal and then into the PC.

Comparing the quality of the G3 (which can act as a USB interface) and the *Rocksmith* Real Tone Cable, I'd have to say I prefer the cable – it sounds just as good (if sometimes a bit quieter), and it's a lot less of a hassle to set up.

BARCODE SCANNING

Anyone with an android phone will probably know that there are a

multitude of barcode/QR code scanners available for Android. However, there is one app that seems especially useful – WiFi Barcode Scanner. This app uses VNC to enable wireless scanning of barcodes to your PC from your phone. For €0.99 in the German store, it's definitely an alternative to look at when contemplating some sort of wireless scanner. Why would you want a barcode scanner? If you're someone who needs to manage a large collection of books, magazines, etc, or want to manage any collection with custom barcodes/QR codes, then you may appreciate the ability to scan barcodes in order to input them into spreadsheets or databases. This is definitely something I'd recommend to anyone thinking of buying a barcode scanner, as it's by



fullcirclemagazine.org

far the cheapest solution I've seen, so long as you already own a camera-equipped Android device. Due to the fact that Google Play offers full refunds (within 15 minutes of downloading an app), it can be a fairly risk-free attempt.

TRANSFER PHOTOS WIRELESSLY

Google has very recently enabled full-size photo uploads to Google+ via instant upload on the mobile app. Originally, it didn't seem like such a useful option to have. However, while writing the HTC One S review (see later in this issue), I needed to take a few photos to demonstrate the camera. Once taken, I would have had to hunt through my room for the micro USB cable in order to transfer them (or else install/use something like Dropbox or AirPlay). Instead, within a few seconds (time will vary depending on your upload speed), they were present and available on my Google+ private album – from which I simply downloaded the photos. So, for anyone who likes quick and easy solutions like this, and if you have a Google+ account, this may be worth noting.



CABLE MANAGEMENT

For the past few years, I've had a constant humming from my (admittedly older) Logitech speakers. Originally, it wasn't a big deal – the humming was quiet enough to be ignored. However, I recently had to replace our router (which sits on my desk, along with all the other electronic stuff I own), and after which my speakers were

humming like crazy, even on the lowest volume setting. I chalked this up to signal interference from power cables, and so I spent a day labelling every cable I had, and then grouping power cables together, ethernet cables together, USB, and audio cables. Once I grouped them, I created a separate “pathway” for each group to run through under my desk. Ethernet cables were wrapped around the desk's supports, power cables ran down the left side of my desk and then along the floor, and the USB and audio cables ran down the right side, spaced a bit apart. The result? No hum whatsoever. I've also noticed fewer disconnection issues via ethernet, meaning I probably had some interference there too. For anyone experiencing similar issues, sorting through your cables may be a very good idea.

I hope at least a few people have found this article useful, and, if you have any questions, comments, or suggestions, you can email me at lswest34@gmail.com. If you do email me, please put “C&C” or “FCM” in the subject line, so that it doesn't get lost in my inbox. I'd be extremely interested to know if any readers find the

occasional music-oriented articles interesting – if they do, I would be happy to do an article on using Audacity to record guitar (electric, acoustic, whatever). If anyone wishes to see some of these articles, just send me an email letting me know.



Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.



HOW-TO

Written by Greg Walters

Programming In Python: Pt 41

Last month, we started our command line version of a library to talk to the TVRAGE web API. This month we will continue adding to that library. If you don't have the code from last month, please get it now from pastebin (<http://pastebin.com/6iw5NQrW>) because we will be adding to that code.

The way we left the code, you would run the program and enter in the terminal window the name of a TV show you want information on. Remember, we used the show Continuum. Once you pressed <Enter>, the program would call the api and search by the name of the show, and then return a list of show names that matches your input. You then would select from the list by entering a number and it would show "ShowID selected was 30789". Now, we will create the code that will use that ShowID to get the series information. One other thing to keep in mind: the display routines are there pretty much to prove the routine works. The ultimate goal here is to create a reusable library that can be used

```
def GetShowInfo(self, showid, debug=0):
    showidstr = str(showid)
    strng = self.GetShowInfoString + self.ApiKey + "&sid=" + showidstr
    urllib.socket.setdefaulttimeout(8)
    usock = urllib.urlopen(strng)
    tree = ET.parse(usock).getroot()
    usock.close()
    dict = {}
```

in something like a GUI program. Feel free to modify the display routines if you want to do more with the standalone capabilities of the library.

The last routine we created in the class was "DisplayShowResult". Right after that, and before the routine "main," is where we will put our next routine. The information that will be returned (there is other information, but we will use only the list below) will be in a dictionary and will contain (if available):

- Show ID
- Show Name
- Show Link
- Origin Country of network
- Number of seasons
- Series image
- Year Started
- Date Started
- Date Ended

- Status
(canceled, returning, current, etc)
- Classification
(scripted, reality, etc)
- Series Summary
- Genre(s)
- Runtime in minutes
- Name of the network that originally aired the show
- Network country
(pretty much the same thing as Origin Country)

- Air time
- Air Day (of week)
- TimeZone

Shown above is the beginning of the code.

You should recognize most of the code from last time. There's really not much changed. Here's more code (shown below).

```
for child in tree:
    if child.tag == 'showid':
        dict['ID'] = child.text
    elif child.tag == 'showname':
        dict['Name'] = child.text
    elif child.tag == 'showlink':
        dict['Link'] = child.text
    elif child.tag == 'origin_country':
        dict['Country'] = child.text
    elif child.tag == 'seasons':
        dict['Seasons'] = child.text
    elif child.tag == 'image':
        dict['Image'] = child.text
    elif child.tag == 'started':
        dict['Started'] = child.text
    elif child.tag == 'startdate':
        dict['StartDate'] = child.text
```



```
elif child.tag == 'ended':
    dict['Ended'] = child.text
elif child.tag == 'status':
    dict['Status'] = child.text
elif child.tag == 'classification':
    dict['Classification'] = child.text
elif child.tag == 'summary':
    dict['Summary'] = child.text
```

As you can see (above), there's nothing really new in this bit of code either, if you've been keeping up with the series. We are using a for loop, checking each tag in the XML file for a specific value. If we find it, we assign it to a dictionary item.

Now things get a bit more complicated. We are going to check for the tag "genres". This has child tags underneath it with the name of "genre". For any given show, there can be multiple genres. We'll have to append the genres to a string as they come up and separate them with a vertical bar and two spaces like this " | " (shown top right).

Now we are pretty much back to "normal" code (shown middle right) that you've already seen. The only thing that's a bit different is the tag "network" which has an attribute "country". We grab the

attribute data by looking for "child.attrib['attributetag']" instead of "child.text".

That's the end of this routine. Now (below) we'll need some way to display the information we worked so hard to get. We'll create a routine called "DisplayShowInfo".

Now, we must update the "main" routine (next page, shown top right) to support our two new routines. I'm giving the entire routine below, but the new code is shown **in black**.

```
elif child.tag == 'genres':
    genrestring = None
    for subelement in child:
        if subelement.tag == 'genre':
            if subelement.text != None:
                if genrestring == None:
                    genrestring = subelement.text
                else:
                    genrestring += " | " + subelement.text
    dict['Genres'] = genrestring
```

```
elif child.tag == 'runtime':
    dict['Runtime'] = child.text
elif child.tag == 'network': # has attribute
    dict['NetworkCountry'] = child.attrib['country']
    dict['Network'] = child.text
elif child.tag == 'airtime':
    dict['Airtime'] = child.text
elif child.tag == 'airday':
    dict['Airday'] = child.text
elif child.tag == 'timezone':
    dict['Timezone'] = child.text
return dict
```

Next page, bottom left, is what the output of "DisplayShowInfo" should look like, assuming you chose "Continuum" as the show.

Please notice that I'm not displaying the time zone information here, but feel free to add it if you wish.

```
def DisplayShowInfo(self,dict):
    print "Show: %s" % dict['Name']
    print "ID: %s Started: %s Ended: %s Start Date: %s Seasons: %s" %
(dict['ID'],dict['Started'],dict['Ended'],dict['StartDate'],dict['Seasons'])
    print "Link: %s" % dict['Link']
    print "Image: %s" % dict['Image']
    print "Country: %s Status: %s Classification: %s" %
(dict['Country'],dict['Status'],dict['Classification'])
    print "Runtime: %s Network: %s Airday: %s Airtime: %s" %
(dict['Runtime'],dict['Network'],dict['Airday'],dict['Airtime'])
    print "Genres: %s" % dict['Genres']
    print "Summary: \n%s" % dict['Summary']
```

HOWTO - PROGRAMMING PYTHON Pt41

Next, we need to work on the episode list routines for the series. The “worker” routine will be called “GetEpisodeList” and will provide the following information...

- Season
- Episode Number
- Season Episode Number (the number of the episode within the season)
- Production Number

- Air Date
- Link
- Title
- Summary
- Rating
- Screen Capture Image of Episode (if available)

Before we start with the code, it would be helpful to revisit what the episode list request to the API

```
ShowID selected was 30789
Show: Continuum
ID: 30789   Started: 2012   Ended: None   Start Date:
May/27/2012   Seasons: 2
Link: http://www.tvrage.com/Continuum
Image: http://images.tvrage.com/shows/31/30789.jpg
Country: CA   Status: Returning Series   Classification:
Scripted
Runtime: 60   Network: Showcase   Airday: Sunday
Airtime: 21:00
Genres: Action | Crime | Drama | Sci-Fi
Summary:
Continuum is a one-hour police drama centered on Kiera
Cameron, a regular cop from 65 years in the future who
finds herself trapped in present day Vancouver. She is
alone, a stranger in a strange land, and has eight of the
most ruthless criminals from the future, known as Liber8,
loose in the city.
```

Lucky for Kiera, through the use of her CMR (cellular memory recall), a futuristic liquid chip technology implanted in her brain, she connects with Alec Sadler, a seventeen-year-old tech genius. When Kiera calls and Alec answers, a very unique partnership begins.

Kiera's first desire is to get "home." But until she figures out a way to do that, she must survive in our time period and use all the resources available to her to track and capture the terrorists before they alter history enough to change the course of the future. After all, what's the point of going back if the future isn't the one you left?

```
def main():
    tr = TvRage()
    #-----
    # Find Series by name
    #-----
    nam = raw_input("Enter Series Name -> ")
    if nam != None:
        sl = tr.FindIdByName(nam)
        which = tr.DisplayShowResult(sl)
        if which == 0:
            sys.exit()
        else:
            option = int(which)-1
            id = sl[option]['ID']
            print "ShowID selected was %s" % id

    #-----
    # Get Show Info
    #-----
    showinfo = tr.GetShowInfo(id)
    #-----
    # Display Show Info
    #-----
    tr.DisplayShowInfo(showinfo)
```

returns. It looks something like that shown on the next page, top right.

The information for each episode is in the “episode” tag – which is a child of “Season” – which is a child of “Episodelist” – which is a child of “Show”. We have to be

careful how we parse this. As with most of our “worker” routines this time, the first few lines (below) are fairly easy to understand by now.

Now we need to look for the “name” and “totalseasons” tags below the “root” tag “Show”. Once we’ve dealt with them, we look for

```
def GetEpisodeList(self, showid, debug=0):
    showidstr = str(showid)
    strng = self.GetEpisodeListString + self.ApiKey
    + "&sid=" + showidstr
    urllib.socket.setdefaulttimeout(8)
    usock = urllib.urlopen(strng)
    tree = ET.parse(usock).getroot()
    usock.close()
    for child in tree:
```

```

if child.tag == 'name':
    ShowName = child.text
elif child.tag == 'totalseasons':
    TotalSeasons = child.text
elif child.tag == 'Episodelist':
    for c in child:
        if c.tag == 'Season':
            dict = {}
            seasonnum = c.attrib['no']
            for el in c:

```

the “Episodelist”, “Season” tags. Notice above that the “Season” tag has an attribute. You might notice (in the code above) that we aren’t including the “Showname” or “Totalseasons” data in the dictionary. We are assigning them to a variable that will be returned at the end of the routine to the

calling code.

Now that we have that portion of the data, we deal with the episode specific information (shown below).

All that’s left now (bottom right) is to append the episode

```

if el.tag == 'episode':
    dict={}
    dict['Season'] = seasonnum
    for ep in el:
        if ep.tag == 'epnum':
            dict['EpisodeNumber'] = ep.text
        elif ep.tag == 'seasonnum':
            dict['SeasonEpisodeNumber'] = ep.text
        elif ep.tag == 'prodnum':
            dict['ProductionNumber'] = ep.text
        elif ep.tag == 'airdate':
            dict['AirDate'] = ep.text
        elif ep.tag == 'link':
            dict['Link'] = ep.text
        elif ep.tag == 'title':
            dict['Title'] = ep.text
        elif ep.tag == 'summary':
            dict['Summary'] = ep.text
        elif ep.tag == 'rating':
            dict['Rating'] = ep.text
        elif ep.tag == 'screenshot':
            dict['Screenshot'] = ep.text

```

```

<Show>
<name>Continuum</name>
<totalseasons>2</totalseasons>
<Episodelist>
<Season no="1">
<episode>
<epnum>1</epnum>
<seasonnum>01</seasonnum>
<prodnum/>
<airdate>2012-05-27</airdate>
<link>
http://www.tvrage.com/Continuum/episodes/1065162187
</link>
<title>A Stitch in Time</title>
<summary>
Inspector Kiera Cameron loses everything she has and finds herself on a new mission when she and eight dangerous terrorists are transported from their time in 2077 back to 2012 during the terrorist’s attempt to escape execution. She takes on a new identity and joins the VPD in order to stop the terrorists’ reign of violence. Along the way, she befriends Alec Sadler, the 17 year old who will one day grow up to create the technology her world is built upon.
</summary>
<rating>8.8</rating>
<screenshot>
http://images.tvrage.com/screenshots/154/30789/1065162187.png
</screenshot>
</episode>

```

specific information (that we’ve put into the dictionary) to our list, and keep going. Once we are done with all the episodes, we return to the calling routine and, as I stated earlier, return three items of data, “ShowName”, “TotalSeasons” and the list of dictionaries.

Next, we need to create our display routine. Again, it’s fairly straightforward. The only thing that you might not recognize is the “if e.has_key(‘keynamehere’)” lines. This is a check to make sure that there is actually data in the “Rating” and “Summary” variables.

```

self.EpisodeItem.append(dict)
return ShowName, TotalSeasons, self.EpisodeItem

```

Some shows don't have this information, so we include the check to make our print-to-screen data a little prettier (shown above right).

All that's left is to update our "main" routine (next page, shown top right). Once again, I'm going to provide the full "main" routine with the newest code **in black bold**.

Now, if you save and run the program, the output of the "GetEpisodeList" and "DisplayEpisodeList" will work. Shown bottom right is a snippet of the Episode information.

That's it for this month. As always, you can find the full source code on pastebin at <http://pastebin.com/kWSEfs2E>. I hope you enjoy playing with the library. There is additional data available from the API that you can include. Please remember, TVRage provides this information for free, so consider donating to them to help their efforts at updating the API and for all their hard work.

I'll see you next time. Enjoy.

```
def DisplayEpisodeList(self, SeriesName, SeasonCount, EpisodeList):
    print "-----"
    print "Series Name: %s" % SeriesName
    print "Total number of seasons: %s" % SeasonCount
    print "Total number of episodes: %d" % len(EpisodeList)
    print "-----"
    for e in EpisodeList:
        print "Season: %s" % e['Season']
        print "    Season Episode Number: %s - Series Episode Number: %s" %
(e['SeasonEpisodeNumber'], e['EpisodeNumber'])
        print "    Title: %s" % e['Title']
        if e.has_key('Rating'):
            print "    Airdate: %s    Rating: %s" % (e['AirDate'], e['Rating'])
        else:
            print "    Airdate: %s    Rating: NONE" % e['AirDate']
        if e.has_key('Summary'):
            print "    Summary: %s" % e['Summary']
        else:
            print "    Summary: NA"
        print "=====
    print "----- End of episode list -----"
```

```
-----
Series Name: Continuum
Total number of seasons: 2
Total number of episodes: 10
-----
Season: 1
    Season Episode Number: 01 - Series Episode Number: 1
    Title: A Stitch in Time
    Airdate: 2012-05-27    Rating: 8.8
    Summary:
Inspector Kiera Cameron loses everything she has and finds herself on a new mission when
she and eight dangerous terrorists are transported from their time in 2077 back to 2012
during the terrorist's attempt to escape execution. She takes on a new identity and
joins the VPD in order to stop the terrorists' reign of violence. Along the way, she
befriends Alec Sadler, the 17 year old who will one day grow up to create the technology
her world is built upon.
=====
```



```
def main():
    tr = TvRage()
    #-----
    # Find Series by name
    #-----
    nam = raw_input("Enter Series Name -> ")
    if nam != None:
        sl = tr.FindIdByName(nam)
        which = tr.DisplayShowResult(sl)
        if which == 0:
            sys.exit()
        else:
            option = int(which)-1
            id = sl[option]['ID']
            print "ShowID selected was %s" % id
    #-----
    # Get Show Info
    #-----
    showinfo = tr.GetShowInfo(id)
    #-----
    # Display Show Info
    #-----
    tr.DisplayShowInfo(showinfo)
    #-----
    # Get Episode List
    #-----
    SeriesName, TotalSeasons, episodelist = tr.GetEpisodeList(id)
    #-----
    # Display Episode List
    #-----
    tr.DisplayEpisodeList(SeriesName, TotalSeasons, episodelist)
    #-----
```



Greg Walters is owner of RainyDay Solutions, LLC, a consulting company in Aurora, Colorado, and has been programming since 1972. He enjoys cooking, hiking, music, and spending time with his family. His website is www.thedesigntedgeek.net.

The screenshot shows the No Starch Press website, which is a catalog of books and ebooks. The header features the No Starch Press logo and the tagline "the finest in geek entertainment". The main content area is divided into several sections:

- Catalog:** A list of categories including Art, Photography, Design, Business, For Kids, General Computing, Hardware and DIY, LEGO, Linux, BSD, Unix, Mac, Manga, Programming, Science & Math, Security, and System Administration.
- New!** A section featuring new releases with book covers and descriptions. Examples include "The Book of GIMP", "Learn You Some Erlang for Great Good!", "Python for Kids", "Master Your Mac", "LEGO Adventure", "The Unofficial LEGO Technic Builder's Guide", "Blender Master Class", "Absolute OpenBSD", "The Modern Web", "Arduino Workshop", "Realm of Racket", and "The BrickGun Book".
- Coming Soon (see all):** A section for upcoming releases.
- Shopping cart:** A section to view the shopping cart.
- User login:** A section with links for "Log in" and "Create account".
- Bestsellers:** A vertical list of popular books with their covers.



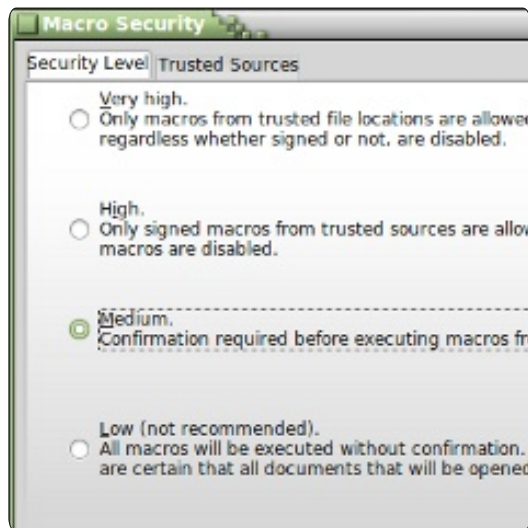
For the previous four parts of this series, we have slowly built a database document using LibreOffice's Base module. We have a database with forms to enter our data, and queries and reports for extracting the data. We now have a usable document for recording our book library. However, our current design has one flaw we need to overcome. If we need to enter a new author or media type while we are in the books form, we have to close the book form and open one of the others. If we could enter new authors and media types directly from the books form, it would behave more like an application and make data entry even easier. We can accomplish this through a few short macros.

The LibreOffice Basic language is very similar to other Basic languages, such as Visual Basic for Applications. To manipulate the underlying LibreOffice document, we access the Uno framework controlling the document. The Uno framework is quite complex, but I will explain, as best I can, the

properties and objects we will use. The goal is not to teach you how to write LibreOffice macros, but how you can use them.

MACRO SECURITY AND OPTIONS

While macros allow us to do cool things in our documents, they can also cause problems. Some people use macros to compromise other people's systems, therefore, we need to take a few minutes to talk about macro security. Whether



you are running LibreOffice on

Linux, Mac, or Windows, malicious code in a macro can compromise your data and possibly your entire system.

Macro security in LibreOffice is simple. Tools > Options opens the Options dialog for LibreOffice. Under LibreOffice, select Security. Click on the Macro Security button to pop up the macro security options. You have four options. Never use the Low security option – it will run macros without asking you. I recommend the Medium security level. With this level, you are prompted whether to run the macros when you open a document containing macros. The High and Very High options require a certificate or folder you designate as trusted. While this is great, I believe nothing trumps the instincts of the user. You usually know whether you were expecting a document to contain macros. When in doubt, click No. Click OK to save your choice and OK to close the options dialog.

Now, on to the fun stuff.

THE MACROS

We will write four macros for our database document. Three will deal with opening forms, and the last will update the list boxes for authors and media types. The general idea behind macros is to accomplish tasks that are not built into the program, or to simplify complex tasks. Our macros really accomplish both, as we will simplify the tasks of adding authors and media types and provide functionality not built into the program.

Before we can begin to write our macros, we need a container to hold them. Macros are contained in a module. Modules can live in the program itself or within a document. Since our macros are specific to our database file, we will embed them in the document. Macros embedded in a document are available only when the document is loaded. Macros contained in the program are available as long as the program is running.

Tools > Macros > Organize
Macros > LibreOffice Basic. The LibreOffice Basic Macros dialog pops up. Select book.odt from the Macro from-list. Click the New button. A dialog pops up asking you for a name for the module. Name it FormCalls. Click OK. This brings up the LibreOffice macro editor. The macro comes with a default main subroutine. We will not use this subroutine. Highlight Sub main and End Sub and press the backspace key to delete them.

Our first macro is a generalized subroutine for opening a form. A generalized subroutine is written for reuse. We will call this routine twice from other routines we write. Type the subroutine shown above into the editor.

The first line of the subroutine is called the signature. The signature determines how the subroutine is called. A signature starts with the keyword Sub, which defines this call as a subroutine. Next, the name of the subroutine. In our case, OpenAForm is the name of the subroutine. Finally in the parenthesis, we have the arguments used when calling this subroutine. In our case, we have a variable named FormName which

```
Sub OpenAForm (FormName as String)
    Dim GetForm as Object
    GetForm = ThisDatabaseDocument.FormDocuments.GetByName (FormName)
    GetForm.Open
End Sub
```

is a type String. In the second line of the subroutine, Dim is another keyword. Dim initializes a variable as a type, and, optionally, a value. We define a variable named GetForm as a type Object. The third line assigns a value to the variable GetForm through a chain of commands in the Uno framework.

ThisDatabaseDocument refers to the currently open database document. In our case, book.odt. FormDocuments is a collection of all the forms in the document. Finally, GetByName retrieves a specific form object from the collection. Notice, we pass the variable FormName from the signature to this method. Once the call is complete, the variable

GetForm is the object of the form name passed to the subroutine. The fourth line calls the Open method of the form. On the fifth line, we tell Basic this is the end of the subroutine with the command End Sub.

We will call the OpenAForm subroutine twice. Once to open the authors form, and once to open the media form. Add the two subroutines shown below to your editor.

The signature on these two subroutines are a little different. Since we will call them from a control within a form, we need to pass the object making the call as an argument, even though we do not use it. The argument oEv is a

reference to the object making the call. We will use this to our advantage later, in the last subroutine, but here we do it because it is required. These two subroutines are pretty simple. We just make a call to OpenAForm passing the name of the form we want to open, Authors or Media.

The final subroutine deals with our problem of refreshing the data in the list boxes for authors and media when we add authors or media using the two subroutines above.

```
Sub ListRefresh(oEv as Object)
    oEv.source.model.Refresh
End Sub
```

Once again, since we will call this subroutine (shown right) from a control, we need a reference to the control making the call. However, this time we will actually use the object. This subroutine makes a method call to the

```
Sub OpenAuthorsForm(oEv As Object)
    OpenAForm ("Authors")
End Sub
```

```
Sub OpenMediaForm(oEv As Object)
    OpenAForm ("Media")
End Sub
```

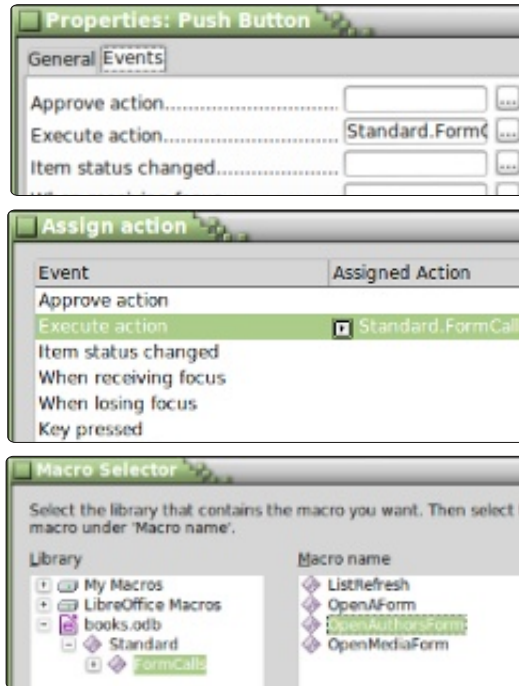

underlying model of the list box and refreshes the data in the list, thus updating our list of authors or media types. Save your module and close the Basic editor.

MAKING CONNECTIONS TO MACROS

At this point, our macros do nothing. We need to connect them to objects in our form to activate them when needed. First, we will connect the open form subroutines to buttons in our form, and then we will connect the ListRefresh to the list boxes.

In the database pane, click on Forms. Right-click the Books form and select edit. Add two push buttons to the form, one under the Authors table and another under the Media table. Right-click the button under the Authors table and select Control to bring up the buttons properties dialog. On the General tab, change the name to AddAuthors and the Label to Add Authors. On the Events tab, click the ellipses (...) button next to Execute Action – which brings up the Assign Action dialog. Click the Macro button to bring up the Macro Selector dialog. In the tree

list under Library, select book.odb > Standard > FormCalls. Select OpenAuthorsForm from the Macro Name list and click OK. Click OK to close the Assign Action dialog. Close the buttons properties dialog.



Do the same with the button under the Media table, only name it AddMedia, make the label Add Media Type, and assign the macro OpenMediaForm to the Execute Action event.

Finally, we need to add the refresh subroutine to our list boxes. Right-click the Authors

column in the authors table and select Column. On the Events tab, click the ellipse (...) button beside "When receiving focus". In the Assign Action button, use the Macro button to assign the ListRefresh macro to the action. This will cause the list to update data from the Authors table when you click on a list box in the column. Do the same for the Media column in the media table. Save your changes to the Books form and close it.

TESTING YOUR CHANGES

Any time we make changes to our forms, we will want to test them and make sure we got everything right, especially in cases where we have used macros. One simple typo could cause things to not work. Double-click the Books form to open it. Add a new book with an author and media type you have not added already. Click the Add Authors button to make sure it opens the form. Add some authors. Close the Authors form. Click on the authors dropdown list box and verify that the authors you added are there. Do the same test with the Add Media Type button and listbox.

FINAL THOUGHTS AND REFERENCES

Again, I would like to emphasize that writing macros in LibreOffice Basic is complex. Documentation is pretty sparse, but it is out there. If you are interested in taking up the challenge, here are some references to get you started:

LibreOffice Basic Guide:

http://wiki.documentfoundation.org/images/d/dd/BasicGuide_OOo3.2.0.odt

Andrew Pitonyak's OpenOffice Macro Information:

<http://www.pitonyak.org/oo.php>

You can find the macros used in this How-To on pastebin.com at <http://pastebin.com/MU2Ztizi>

Next time, we will move on to another part of the LibreOffice suite and explore the Math module.



Elmer Perry's history of working, and programming, computers involves an Apple IIE, adding some Amiga, a generous helping of DOS and Windows, a dash of Unix, and blend well with Linux and Ubuntu.



HOW-TO

Written by Jeremy Boden

This is very easy to set up – although, later, I shall use a slightly complicated example to illustrate its use.

If you have several users configured on your machine, each user gets their own independent schedule which they can configure.

Experienced users will recognise this as a description of Cron, pre-installed on virtually all Linux machines. The use of cron by root is somewhat different, and is used by some system functions – so we shall ignore the use of cron by root.

Each user specifies their schedule via a “crontab” which can be listed or edited via the crontab command. We can list our crontab with the crontab -l command:

You will always find these descriptive comments in your crontab – and you should retain them. To customise your crontab you need to add one (or more) lines containing time information and a command (or script) to be

run.

Cron runs as a background daemon, and will run your command(s) at the appropriate times.

Note: your commands will be run under your user privileges; it isn't possible to gain elevated privileges by using sudo, etc.

There are 5 fields to specify the date – and the formats can be made quite complicated.

The separator between each of the five fields is always one or more spaces (or tab characters). Individual fields may contain complicated specifications such as 1-5,10-15 (but are usually either an asterisk or a single number). The hyphens and comma are not to be considered as field separators.

Each one is typically a number of units. So, for example, 2,4,6 means run at 2 units, 4 units, and 6 units. Another variation is 2-5 which is the same as 2,3,4,5. If we want to run on every valid

occasion, we should specify a *. Writing something like */10 means run once every 10 units (so */10 in the minutes field means run at 00,10,20,30,40,50 minutes past the hour).

We may also use names in those fields where it makes sense.

As if this wasn't enough, combinations are allowed e.g: 1-3,7,9 is equivalent to: 1,2,3,7,9

1st field
Minute of the hour (0 - 59)

2nd field
Hour of the day (0 - 23)

3rd field
Day of the month (1 – 31)

4th field
Month of the year (1 – 12 or feb, jun, etc)

5th field
Day of the week (0 – 7; both 0 and 7 mean Sunday, or names: mon, tue, wed, etc).

```

jeremy@HECTOR:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command

```

EXAMPLES

First, a bad example – this would run only if 13th June is a Monday (next occurs in 2016)!

```
0 1 13 jun mon some-command
```

More sensibly,

```
0 1 13 * * some-command
```

This would run at 01:00 on the 13th of each month.

Suppose you want to log problems with a very bad connection, using a user-written script named `.whatip.sh` which is a (hidden) file in the home directory, you might run the following:

```
*/10 * * * * [ -x  
.whatip.sh ] && bash  
.whatip.sh 2>/dev/null
```

This runs every 10 minutes. The `[-x .whatip.sh]` command tests that the executable file exists, and only if this is true does the `&&` permit running of the script (which writes to a log file); the `2>/dev/null` causes any output to be ignored.

Despite the wealth of possibilities in specifying the time a job is to be run, there are certain

specifications which can't be easily specified, such as "run on the last day of the month". For such cases a run specification of something like:

```
0 1 28-31 * * some-script
```

would be suitable – with the script making an early exit if it isn't actually the very last day of the month.

EDITING CRONTAB: USING THE CRONTAB -E COMMAND.

This will almost certainly not use your normal editor, such as `gedit`, but a more basic one – such as `nano`. This is perfectly good for editing configuration files and works in a non-GUI environment.

You still have keys such as insert, delete, backspace. Screen navigation (up, down, etc) is via cursor keys and page-up, page-down keys. However, you don't have scroll bars of any sort.

The function keys are listed across the bottom – but you need only two of those listed.

Scroll down to the last line and enter your changes to your

`crontab`.

I understand that the very last line must end with a newline – this may not be necessary.

Check, then press CTRL+O to write out your changes.

Finally, press CTRL+X to exit.

If you want to disable a `crontab` specification, comment it out by inserting a `#` in the first position.

It's quite hard to come up with a `crontab` line which does useful work, without turning it into a script – but I do have a useful example (the credit for this belongs to an unknown author).

First the problem: If you move a fair number of files around, then Nautilus will create a small thumbnail file for it. If you move the file, you get another thumbnail, and if you look at your system directories you will generate many thousands of thumbnails. The problem is that Nautilus never deletes thumbnails.

To check if you have a problem, enter the following command:

```
du -sbh .thumbnails
```

I get a value of 20M i.e. 20MBytes (that's roughly 1,000 thumbnails). You might see a much larger value.

The solution: Every day I run a cron job to delete thumbnails which were last accessed more than 7 days ago. The command part in the `crontab` is:

```
find ~/.thumbnails -type f  
-atime +7 -exec rm '{}' \;
```

It is critical that the command is entered exactly as shown including the trailing `\;`

Because this command includes the `rm` (remove file) command, you may like to test it first by running another closely associated (and harmless) command in a terminal window:

```
find ~/.thumbnails -type f  
-atime +7 -exec ls '{}' \; |  
more
```

Note the replacement of the `rm` command by the `ls` command.

Once you are happy with the operation, you can enter a time specification and the command

containing the rm operation into your crontab. Obviously you may want to change the value of +7 to suit your machine. My crontab entry reads:

```
45 19 * * * find
~/.thumbnails -type f -atime
+7 -exec rm '{}' \;
```

Or, each evening, at 19:45 remove excess thumbnails.

If you have multiple users, you will need to repeat for each one.

Your crontab is actually stored in a sub-directory of the /var directory – so a system upgrade where you choose to replace all your system files (even if you retain your /home directory) will cause your crontabs to be lost. However, it is important that you edit your crontab only via the crontab command, as this incorporates some important error checking.

EXCURSIONARY NOTE

If you have not performed an install which involved overwriting your /home directory in the last year or two, then the thumbnail removal entry may work slightly

differently – but the difference is marginal and rather historical.

Originally, in addition to setting creation and modification timestamps on files, Linux always recorded file access timestamps (this is the -atime in our crontab entry). This can be inefficient, since reading a file always caused an additional write (to update the access timestamp). It is now possible to indicate that you don't want any atime updates to take place – and this is the current default.

Note that frequently accessed directories would always be looked at more often than every 7 days and so (under the old scheme), their thumbnails would never be deleted. Under the new scheme, atime is never updated, and so even these thumbnails would be deleted and very quickly recreated every 7 days. It's not a noticeable problem for thumbnails – although I understand one or two older applications find the atime change to be a problem.

To check your setting enter:

```
cat /etc/fstab | grep /home
```

into a terminal screen; I get 2 lines displayed.

```
# /home was on /dev/sda7
during installation
```

```
UUID=0648d2d1-9a41-4257-8b79-
dfc7bc227e82 /home ext3
defaults,noatime 0 2
```

(My /home is mounted as ext3 – yours is probably ext4).

I have manually changed /etc/fstab to mount everything as noatime – which means no access timestamps are updated, just like the modern default. If you don't see noatime listed, or you see atime in its place, you should consider modifying your fstab file to use the noatime option unless there is a special reason not to.



The Ubuntu Podcast covers all the latest news and issues facing Ubuntu Linux users and Free Software fans in general. The show appeals to the newest user and the oldest coder. Our discussions cover the development of Ubuntu but aren't overly technical. We are lucky enough to have some great guests on the show, telling us first hand about the latest exciting developments they are working on, in a way that we can all understand! We also talk about the Ubuntu community and what it gets up to.

The show is presented by members of the UK's Ubuntu Linux community. Because it is covered by the Ubuntu Code of Conduct it is suitable for all.

The show is broadcast live every fortnight on a Tuesday evening (British time) and is available for download the following day.

podcast.ubuntu-uk.org



HOW-TO

Written by Nicholas Kopakakis

Blender - Part 3

Many, many times in your blender sessions you will press the TAB key. It's the keyboard shortcut for switching to Edit Mode.

Load the snowman.blend file that we created last month, and, on the header, locate the button showing that you are in Object Mode (it's called header, but it is placed by default where a footer should be).



Select with the RMB (Right Mouse Button) the head (top UVsphere) of our snowman.

Press the TAB key. The entire header changes and now you are reading "Edit Mode" on the same button. (You can alternatively press that button, and, from there, you can select the edit mode. For now, we don't need to know what the other modes are).



Also the UVsphere changes to a orange-wired sphere.

You are in edit mode. While in edit mode, you are able to adjust and manipulate only the selected object. Try, for example to select with the RMB the lamp or the nose. You can't. You can select only a vertex or group of vertices of the object that you are in.

Vertex is a 3-dimensional coordinate,

Edge is a line connecting two vertices and

Face is the planar field in between 3 or more vertices.

In edit mode, you can select vertices, edges and faces – the same way we saw in the first part of this series, 2 months ago. Use the box selection tool, the lasso tool, the circle selection tool, RMB and Shift+RMB to select multiple vertices.

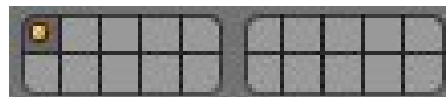


Also in the header, observe three small buttons representing the vertex, edge and face selection.

From these buttons you can change what you select (vertex, edge or face). Alternatively, press the keyboard shortcut Ctrl+Tab.

Go ahead and switch back to object mode by pressing the TAB key.

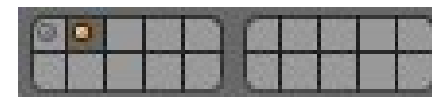
On the header again, you can see that the buttons that we mentioned before have disappeared, and some others came back. Let's focus for a minute on the buttons showing below



These are called Layers Buttons. The yellow dot in the upper left button indicates that we have objects in that layer selected, or recently selected, and the darker gray color that we are currently in that layer.

Press the number 2 on your keyboard (NOT the numpad) or the second button from the upper left corner. Everything disappears – as we are now on a different layer.

Press Shift+S for Snap menu, and select Cursor to Center. Press Shift+A to add a new object. From the Mesh group select Cylinder. Notice now the Layers Buttons.



It shows us that we have objects in two layers, and the second layer is the active one with something selected. Press S for scale, and 0.3 to scale it down 30%. Press G for grab, Z to lock it to Z-axis, and 0.3 to move your cylinder up 30% of a blender unit.

If it's not selected already, press Numpad-5 to switch to orthographic view (explained in issue #69) and Numpad-7 for top view.

Press the TAB key to enter the edit mode. Press Shift+S and Cursor to Center to move your cursor again to the center of your grid – in case you have moved your cursor. Press Shift+A to add a Circle from the Mesh group. After that, press F6, and, from the window

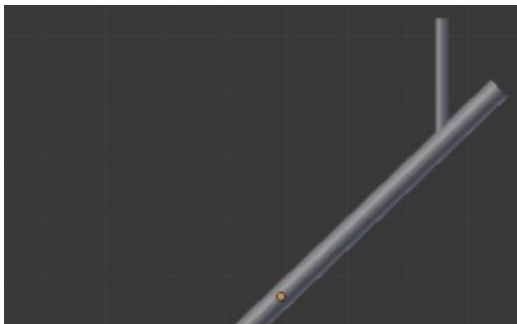
HOWTO - BLENDER Pt 3

that appears, change the Fill Type from Nothing to Ngon. After that, press S for scale, and 0.5 to scale down your circle by 50%.

Press the TAB key to return to object mode. While in edit mode, the objects that you add (in this example the circle) are all parts of a single object and blender manipulates them as a single object.

We have the hat. Now let's create the hands of our snowman. Press 3 on your keyboard (NOT Numpad-3) to work in Layer number 3.

Shift+S again to move our cursor to center. Shift+A to add a new Mesh. A cylinder again. Press S for scale, Shift+Z to keep the dimensions of Z-axis, and alter the other 2 axis, X and Y. Press 0.02 for scaling 2%. After that, press S again, Z to alter only the Z-axis



dimensions, and 0.5 to make it half long. Lets go to edit mode again. With the cylinder selected, press TAB. Add a new cylinder, scale it down, rotate it, and move it in order to create a branch arm.

When you are done with your branch, Press Alt+1 on your keyboard (NOT Numpad-1) to select Layer Number 11.

TIP: Press 1, 2, 3... to 0 on your keyboard: you select your active layer from 1 to 10. This is the upper row on your Layers Buttons mentioned earlier. Pressing Alt+1, Alt+2... to Alt+0 selects the lower row of layers from 11 to 20.

With Layer 11 active, add a new Cylinder (by now you have to know how to add a new Mesh).

Press TAB to switch to edit mode. Press Z key to display your cylinder as wireframe. This is a very useful view of your object, and I personally use it a lot to toggle the view from solid to wireframe. Press A to select all vertices (if not already selected). (When in object mode, pressing the A key selects all objects, but, when in edit mode, this key selects or deselects all vertices.)

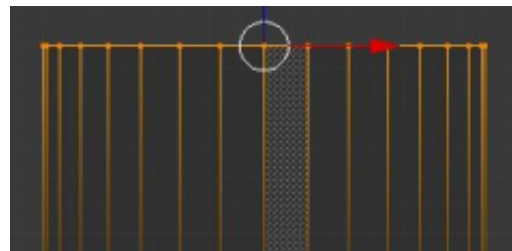
Now it's time to create a pipe.



Assuming that you have a front orthographic view, and the small button on the right of the mode selection button indicates the wireframe symbol...select all vertices.

Scale down your cylinder to 8%.

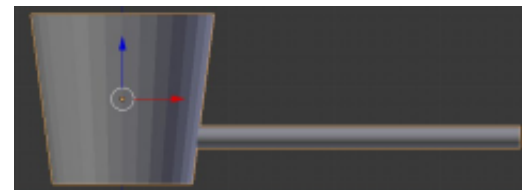
Press A to deselect all vertices. Now, box select (using the B key) only the upper vertices.



Scale up a little by pressing S key and moving your mouse away from the center of your selection. Press A to deselect all vertices, and box select the lower vertices only.



Scale down a little by pressing S key and moving your mouse towards the center of your selection. After that, add a new cylinder, scale it down, and create the rest of your pipe.



It is a good time also to replace that nose that we did last month (using a cone) with a new one based on a cylinder – because we don't actually want a so sharp edge for our carrot-nose.

Now we have almost what we need.

Do the same as we did with the cylinder a little earlier to create another branch arm. Place it in a new layer or in the layer where you have the other branch. You can also duplicate your branch arm by selecting it and pressing Shift+D.

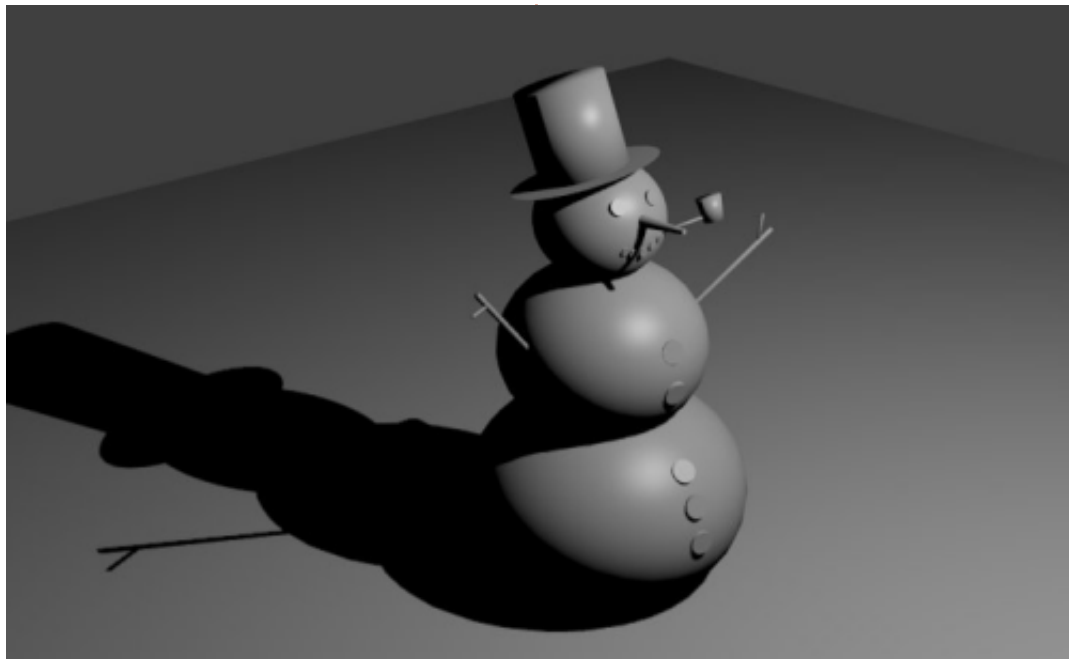
Now that you have all the elements in your layers, you can either move all objects to the starting layer (Layer 1), or by pressing Shift and the Layer-

buttons that have objects in them – to combine the view of all layers that are selected.

TIP: To move an object to another layer, select the object and press M key. From the Move to Layer window, select where you want to move your object.

Next month, we will add some color in our lives, introducing textures!

For this month, I'll suggest <http://www.blenderartists.org> – a forum where you can find the latest news, job ads, artworks, and much more blender-related stuff.



Also for this month, I have a video for you. Go to http://www.youtube.com/watch?v=USyoT_Ha_bA, and have a look back in 1963 at the ancestor of all computer graphics, Ivan Sutherland's Sketchpad (my Wacom tablet almost cries).



Nicholas lives and works in Greece. He is working for a post-production house (commercials - films) for several years. Three months ago he migrated to Ubuntu because "it renders faster". Blender found him two years ago.

CODEWORD

Every number in the grid is 'code' for a letter of the alphabet. Thus the number '2' may correspond to the letter 'L', for instance. All - except the difficult codeword puzzles - come with a few letters to start you off

10	13	26	15	17	22	18	25		13	9	22	5
	8		9		18		16		6		13	
19	16	24	16	25	1		15	2	22	26	16	6
	14		25		18	5	18		18		25	
4	17	25	15		13		6	2	24	20	18	8
	2				15				13			
25	15	2	2	8	25		26	13	8	15	22	5
			11				22				18	
10	22	13	3	13	15		17		19	13	10	15
	2		16		2	8	6		8		16	
17	26	22	2	13	22		18	7	13	10	15	25
	18		17		18		24		16		18	
13	25	21	25		25	13	15	16	22	16	25	15
1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z												

Solutions are on the second last page.

Puzzles are copyright, and kindly provided by,
The Puzzle Club - www.thepuzzleclub.com



HOW-TO

Written by Mark Crutch

Inkscape - Part 10

The previous instalment included a comic strip which was made in Inkscape using paths, ellipses and rectangles with flat fills and gradients – all elements that have been covered in this series so far. But it also included one other type of object which is an essential element of many images: text.

Creating text objects in Inkscape isn't difficult, but does come with a few caveats that can easily trip up beginners. Some of these are common to all vector graphics programs, but the first issue you're likely to face is peculiar to Inkscape and involves a brief history lesson...

Inkscape's native file format is SVG, an open format specified by the World Wide Web Consortium (W3C). By using an open format, Inkscape creates files that can be viewed and edited, at least to some extent, in a wide range of applications. This is a huge benefit to the user, whose files aren't locked-in to being used just within Inkscape, but has the disadvantage

that the Inkscape developers have little direct control over what makes it into the specification, and what doesn't.

The first version of the SVG spec was released in 2001, followed by version 1.1 in 2003. After that, several years were spent working on version 1.2, which was to include many additions and improvements – including additions to support text which will wrap and reflow to fill its container. The Inkscape developers spent quite some time implementing support for this “flowed text” format, fully anticipating its official release in the new SVG standard.

Then SVG 1.2 withered and died. It never became a standard and to this day – 8 years later! – SVG 1.1 is still the latest official version of the SVG specification. This left Inkscape with the ability to create objects that are compatible with only an aborted spec, but, as this facility had made it into a release version of the software, it would break

compatibility with users' files if the code was simply excised. The Inkscape developers took the pragmatic decision to leave the Flowed Text feature in place, even though using it will create files that other applications will not fully understand.

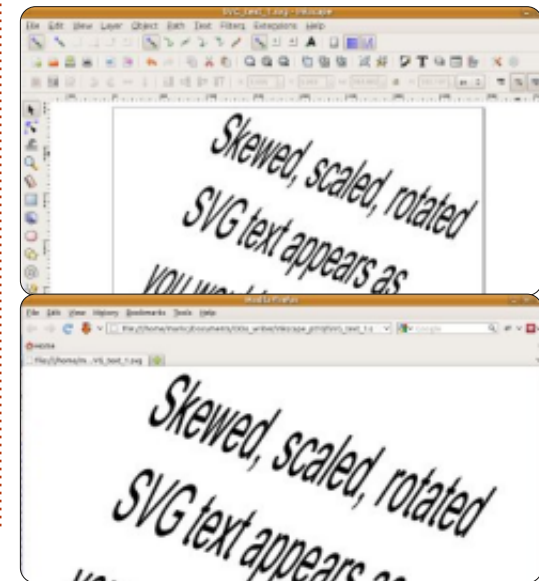
As a result of this historical issue, Inkscape can create text in two different forms: the SVG 1.1 type, which doesn't automatically flow into its container (which I'll be referring to as SVG Text), and the SVG 1.2 type which does flow (Flowed Text), but which doesn't conform to the SVG specification. The problem arises because it's far too easy to accidentally create Flowed Text, especially if you have previous experience with other graphics applications.



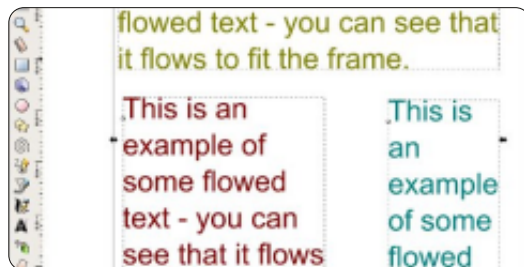
Let's get practical and actually create some text. First select the Text tool by clicking its icon in the tool palette, or by pressing “T” or F8.

Now, simply click inside the drawing window and start typing.

If you can't see anything, check that your color and opacity settings make sense via the status bar. Congratulations, you've just created some SVG Text. If you switch back to the Select tool using the tool palette or F1 key, you can move, scale, skew and rotate your text object in the same way as any other SVG element. Because this type of text object conforms to SVG 1.1, it can be displayed or edited by various other applications. As you can see from this image, even skewing and rotating the text object in Inkscape (below) isn't enough to prevent it displaying in Firefox (bottom):



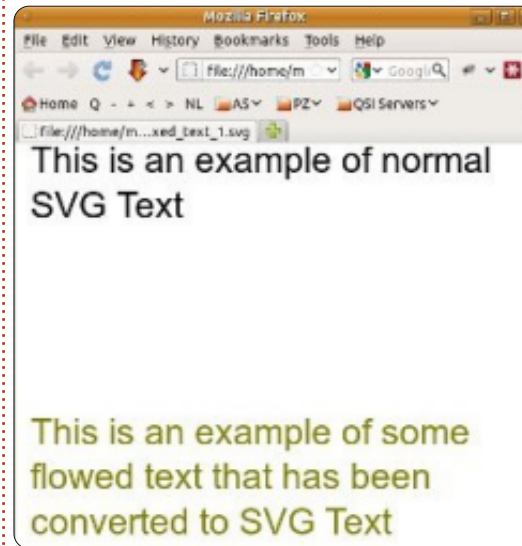
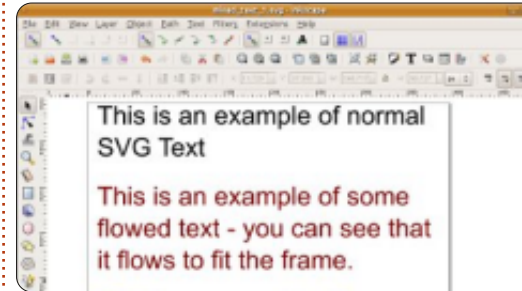
Some other graphics applications require you to drag a rectangle on the canvas to contain your text. This is especially common in desktop publishing programs such as Scribus, where almost everything is defined by drawing a frame to contain it. You can do this in Inkscape as well – just select the Text tool then click and drag a rectangle onto the canvas before typing. You've now created a Flowed Text object. With the Text tool still active, and the Flowed Text object selected, you should see a small square handle at the bottom-left of the text frame. By moving this handle you can change the size and shape of your frame, and the text will re-flow automatically. The following image shows the same Flowed Text object duplicated a couple of times in Inkscape. The copies have had their frame sizes changed, and you can easily see that the text has moved around, and, in the case of the bottom-right frame, it has been automatically truncated:



If we load this SVG file into Firefox, the result is a blank page. Firefox ignores the Flowed Text completely, and the same applies to almost every other application. Remember, the difference between creating SVG Text and Flowed Text is as simple as whether you just click, or click-drag. If you want to use your SVG files in other applications, you should almost always just click when creating your text objects. If you're in any doubt, select your text object and check the status bar, which will describe the object as either "Text" or "Flowed text".

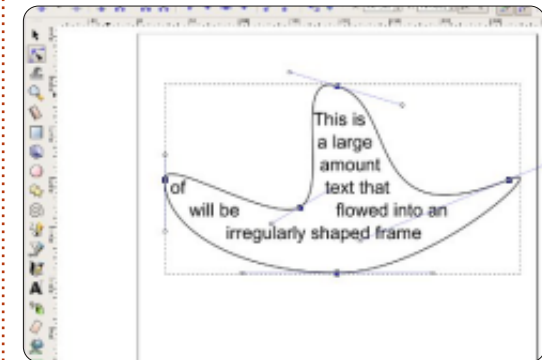
Despite the tone of the previous few paragraphs, there are sometimes valid reasons why you would want to use Flowed Text. If you don't want to use your Inkscape files in another application, then the presence of non-standard SVG code won't affect you. Even if you do want to use your SVG files elsewhere, it can sometimes be easier to create Flowed Text during the drawing stage, and then convert it to SVG text using the Text > Convert to Text menu entry just before you save the final version of your file. Loading the file into Firefox gives

exactly the result you would expect:



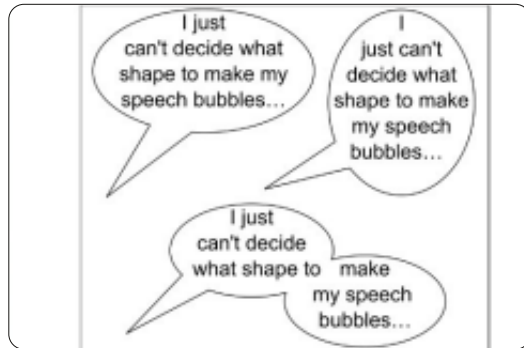
The real advantage of Flowed Text in Inkscape is that it can flow into shapes other than simple rectangles. First you will need a shape for the text to flow into: this can be any of the simple Inkscape primitives such as rectangles, ellipses and stars, or it can be a path element which allows you to create complex shapes using all the Boolean operations and node editing tools that have been

described in previous articles. It must be a single object though, so can't be a group. You will also need some text, but it doesn't matter whether you create SVG Text or Flowed Text at this stage. Select both your shape and the text, and then use the Text > Flow into Frame menu to perform the magic. Note that the status bar now describes your text object as "Linked flowed text", and that you can modify your shape as much as you like, with the text re-flowing to fit:



As with normal Flowed Text, this will not be understood by other SVG applications. You can still use Text > Convert to Text, although with very complex paths you may find that the text shifts around a lot during the conversion. Nevertheless, for labels and speech bubbles, the ability to change your container shapes and have the text re-flow to suit, can

be a real time saver.



Now that you know how to create basic text objects in their various forms, it's time to exert a little more control over the style of your words. Most commonly, you'll want to choose a suitable font, set its size, and perhaps change the justification. All of these options are available from the Tool Control Bar, and, although they can be changed at any time, it's often easier to set them before you click (or click-drag) to place your text cursor to avoid problems with the focus being in the wrong place when you start to type.

The drop down menu to the left of the toolbar lets you select a font. Inkscape can be a little fussy about its fonts, so you may find that some fonts on your system aren't available, especially those that haven't been created by a professional type foundry. There's also an occasional glitch that can



occur when you first open this menu: Inkscape shows a shortened version of it, with just a few fonts listed. If that happens, simply click away from the menu to close it, and then re-open it. Finally, you may find that some fonts simply refuse to stay selected when picked from this menu. I commonly have this problem with "Arial Black", which Inkscape immediately replaces with a bold version "Arial". In this case, you can use the "Text and Font" dialog from the Text menu to select the stubborn typeface, which will be used when you click the Apply button.

The size drop-down lets you pick from a few predefined sizes, all in units of "SVG pixels". You can also type directly into this box to specify a different size, but there's no way to use any other units. The following two buttons are toggles, used to switch to bold or italic versions of the selected font – or a bold-italic version if both are active.

Finally, there are four buttons for setting the text's justification. Left, centre and right justification

can be used for any piece of text, but full justification (where the program tries to line up both the start and end of every line) is available only for flowed text. You can convert fully justified flowed text into SVG text – which does a surprisingly good job of maintaining the justification, but it plays havoc with Inkscape's on-page text editing if you need to subsequently change the content.

Having created your text objects in Inkscape, you may want to transfer the SVG file to another machine, or post it online. In doing so, there's a good chance that you'll run into a problem with missing fonts. Text in SVG files is stored as a string of characters, together with some style information which contains the font name. If an identically named font isn't present on the destination computer, the software used to display the file will substitute it with an alternative, often with dramatic effects on the appearance of your image. The image right shows a couple of panels from one of my comic strips, first as it should

appear using a couple of commercial comic fonts, and then how it appears on an Ubuntu box that doesn't have those fonts installed.

There are four possible solutions to this problem:

- Ensure that the required fonts are present also on the destination machine. This may not be possible if it's not your computer, or if the font licence prohibits it.



- Don't use the SVG format to transfer files to other systems. For my comics, I always export a PNG version of the file to ensure that my readers see a pixel-perfect representation of the image. Exporting to other formats will be the subject of part 12 of this series.
- Only use commonly available fonts. This is not a perfect solution, but may be viable in some circumstances. My "Greys" comics, for which I make the SVG files available to download, use the ubiquitous "Arial" font for this reason. Even with that precaution, they often display poorly in a web browser, and although this approach makes it easier to open the files in Inkscape, I still produce PNG versions for the benefit of casual readers.
- Convert your text into paths before saving the file.

That last solution is as simple as selecting the text then the Path > Object to Path menu entry. In Inkscape 0.48, your text will be replaced with some identical looking paths, one for each letter, grouped together. Earlier versions simply produced a single composite path that contained all the letters. Once converted to

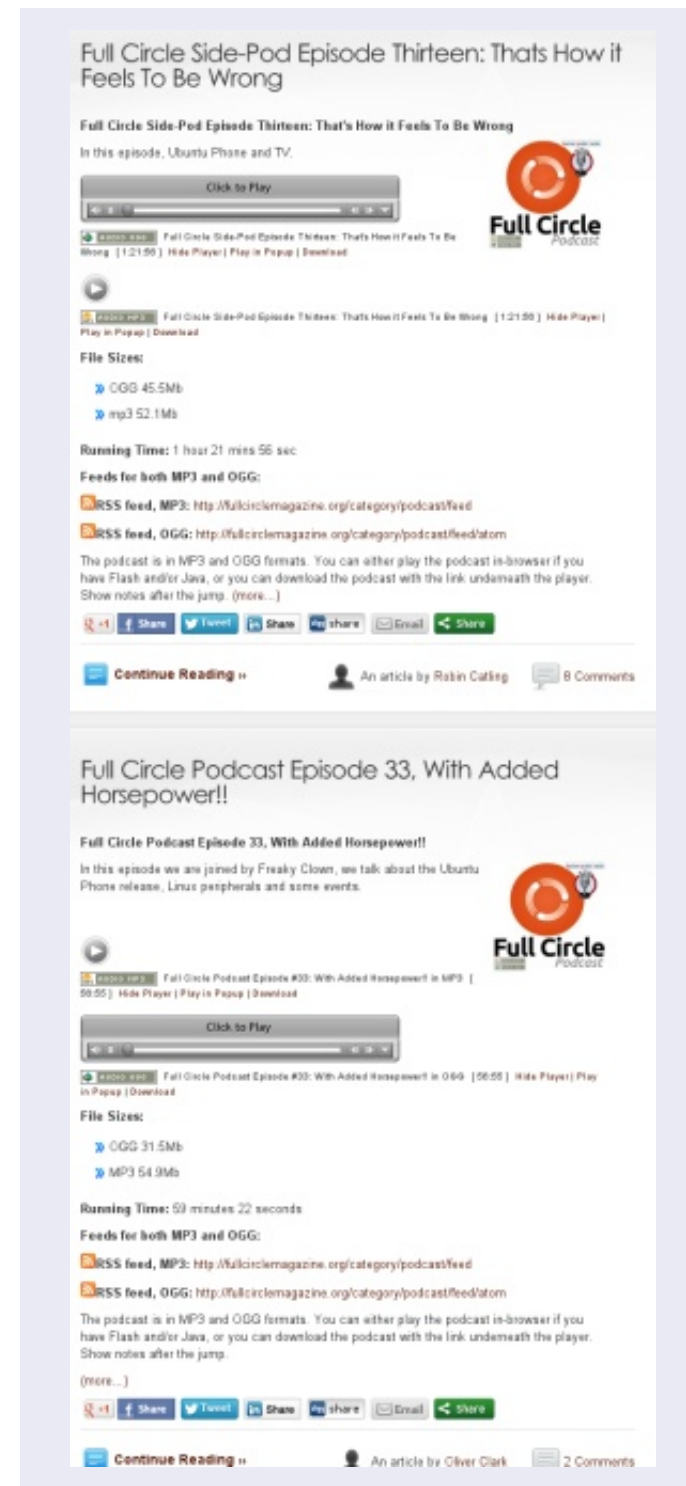
paths, the text is just another collection of shapes in your SVG file and no longer require the fonts to be installed.

This might sound like an ideal solution, but does come with a drawback: your "text" is no longer a text object, so can no longer be edited using the text Tool in Inkscape. If you decide to use this approach I recommend performing the conversion to paths as late as possible. You should also make a copy of the text object before you do so, in case you need to edit it again at a later date. The text object should then be put onto a hidden layer, dropped behind some other object, made transparent, or secreted away in some other manner so that it doesn't interfere with the rendering of the file on the destination machine.



Mark has been using Linux since 1994, and uses Inkscape to create two webcomics, 'The Greys' and 'Monsters, Inked' which can both be found at:

<http://www.peppertop.com/>





This month, we are going to talk about localStorage. This localStorage is built into the browsers for you to use to hold information that you can retrieve later. With that said, other websites can also use it. When you are writing your code, you should write it with that in mind.

localStorage is a pretty powerful tool if you learn how to use it properly. It can allow you to give the user a place to hold data without holding it on your website. localStorage works by using key/value pairs. This basically means this = that, key = value, or itemName = itemInfo. The value can hold only strings of information, but there are ways to get around that if you want to hold a set of strings. That is what we will be doing.

Check out your own localStorage, just run a `console.log(localStorage);`. There are other ways of checking your localStorage, however they are not very reliable. Doing it this way, you know you are going to have your

answer returned to you.

We are going to make a lot of changes to your js file to adapt it for use of local storage to hold our Ubuntu Versions. Since we will be doing so many changes, we will go through the whole main.js file with changes already added.

Before we go into the main.js file, we are going to add a button in the index.html file. I added my button after the form section, and before the list. Here is that code:

```
<section>
    <button
      onClick='clearLocalStorage()' >
    >Clear localStorage</button>
</section>
```

This button is pretty simple. The onClick function, as an attribute, will tell javascript to run whatever is in the attribute. The text between the button tag will be what is displayed inside the button.

Now that we have a “Clear

localStorage” button, let’s move on to the fun JavaScript. Starting from the beginning, we have a console log to tell us that our js file is connected. If the functions do not contain anything new, I will just be mentioning them with their code.

```
console.log('js connected');
```

Next we have our get element function “ge”, so that we can easily grab elements from the DOM by their ID.

```
function ge(id) {
    var theElement =
document.getElementById(id);
    return theElement;
};
```

The processForm function changes a little bit:

<http://pastebin.com/HuLiH3tw>

First in the processForm function, we are preventing the default behavior of the form and console logging, to let us know what is going on in the js as it

happens. The next console log is making sure we know what our elements are and what we can do with them.

This next portion of the function is checking to make sure our form fields are not blank, and letting the user know if they are. If they are not blank, then we are calling a function to store the data, and then display the data. The display data function will be a new function that we can call on anytime something happens to the data.

After we are done working with the data, we are going to clear the fields using `ge('id').value = ''`. Last but not least, we make the function return false to make sure the default behavior is prevented.

The storeData function also has had a few changes made to it:

<http://pastebin.com/ef2Q7py4>

The first thing we are going to do is create a key. This if statement is asking if the submit button has a key attached to it:

- If it doesn't, it will create an ID. The ID it is creating will always start with ubuVers. This is so we can make sure the data we are about to pull back in and display, actually belongs to us. The second part of the key is a random number multiplied by 10000001. This way, if the random number comes back as 3, it will still have a long enough string of numbers to be unique.
- If it does have a submit button, it will set the ID to that key. That sets us up for editing the versions in our list if we need to. We will see that action a little bit later.

The next portion of this function is giving variable names to the form elements, and adding their values into a dictionary, nothing too special there. After we console log those values to make sure we are doing what we think we are doing, we have a very special line of code.

"localStorage.setItem(id, JSON.stringify(ubuVersDict));" is a line that is actually setting information to localStorage. If the ID already exists in storage, it will overwrite the information (hints the editing), and, if not, it will make a new entry.

The second bit of information

getting passed to setItem() is the data we put the version and release into. The data type for that is known as many things, but, in JavaScript's case, it is called json. Using the JSON object, we are going to "stringify" the information going into the localStorage. This means that it is creating a string out of the json object. This is important if you want to be able to access that data later, otherwise it will go in as [object Object].

Last up in the function are two lines that can make your life very difficult if you forget them. The first is to console log the localStorage object so debugging is easier if we need to. The second line is to make sure the submit button has a value of "Add". The "Add" value on the submit button is another one of those things that goes with editing data later.

The displayData function is a very important piece of this puzzle. As I told you earlier, this will fire every time something happens to our data. Let's take a look at that now:

<http://pastebin.com/TPqg31ye>

The first thing we are doing in



this function is to clear out everything in our list. innerHTML is a great function if you need to clear or set information inside of an HTML element. Immediately after clearing our list, we are going to jump into going through the localStorage data.

First up is to check if it is empty:

- If it is, we are going to let the user know that there is nothing being stored. Otherwise, we are going to look inside it. localStorage.length is counting how many items are within the array; if it is 0, it is empty.
- If it is not empty, we have the opportunity to go and check things out. The first thing we do is loop through all of the items. We will need to set a variable for the key and the object inside the value. These variables are very important to the whole function and the whole project. The key variable is simply grabbing the key from local storage by using the key object (localStorage.key(i)). The i being passed to the key object is the integer that is being used to count through the for loop.

The value variable is passing the key variable through the getItem object, which returns the

information stored in localStorage for the given key. The third variable, obj, is changing the stringified information back to JSON data. JSON.parse will change any stringified JSON data back to valid JSON we can access again.

The next if statement is checking to see if the key is one of our "ubuVers" keys. Using substring(x,y) is a way to grab a part of a string. x being the character you would like to start with, and y being the character you would like to end on. The ending character is not included in the grabbed substring. For example, substring (0,2) of the word cat, will return "ca". the 0 character being c, and the 2 being t.

If the key does start with "ubuVers", it is one of ours, and we need to process it. The first thing we are going to do is console log the key and value out for debugging, then we are going to make a whole bunch of variables.

The first variable in this list is the list element that we have been writing our versions to. Next we are creating an li element to hold our ubuntu version. Still nothing new here. The third, fourth, and

fifth items on the variable list are grabbing our version and release information and formatting it for our list. obj was the variable we made earlier to hold the value from our localStorage information and inside of that was a release and a version. obj.version and obj.release is how we get that information back out. And last but certainly not least:

itemDeleteButton,
itemDeleteButtonText,
itemEditButton, and
itemEditButtonText are creating elements and text for our new edit and delete buttons.

Now we are going to jump into setting attributes and onClick functions for new buttons. Each button needs to be created per item so they can be attached to the item it will be editing/deleting. Each button is getting a new attribute with setAttribute. The attribute is ID, and is being set to its key. Then each button is getting an onclick command to interact with javascript when pressed. The delete button is running a function that will be calling a deleteItem() function with an ID being passed to it. The edit button is doing the same with a respective editItem() function.

```
function deleteItem(key) {  
  // as to make sure the user wants to delete this  
  if (window.confirm('Are you sure you would like to delete this version?')) {  
  
    // remove the item from localStorage  
    localStorage.removeItem(key);  
  
    // let the user know it happened  
    alert('The version was deleted successfully');  
  
    // reload the view  
    displayData();  
  }  
}
```

listItem.innerHTML and the respective edit and delete buttons are setting up the elements' HTML between the tags. In this case it is just text, but could be full out HTML if you wanted it to be. Which finally brings us to listItem.appendChild() commands. These commands are adding the edit button to the list item, then delete, and finally adding and displaying the list item in our list.

The next function we have now (shown above) is a deleteItem function that we used on our delete button. This function is fairly straightforward.

The first thing we want to do is confirm that the user would like to really delete this item, so we ask them with a window.confirm()

prompt. This prompt gives a cancel or ok option. If OK is pressed, it will come back into your JS as true; if cancel, it will come back as false. So if (true) { do this } else { nothing here }. If they do in fact want to delete the item, we simply pass the key that was supplied to the function into localStorage.removeItem(); and it is gone. We then let the user know it was successfully deleted and we

refresh our list.

The editItem function (shown below) is almost as simple. As I mentioned earlier a couple times in other functions, we already set those up for editing. Here is the goodness that makes all that work.

In this case, we are grabbing the information from localStorage

```
function editItem(key) {  
  // get the object being edited from local storage  
  var value = localStorage.getItem(key),  
      obj = JSON.parse(value);  
  
  // populate the form with the object  
  ge('ubuVersNum').value = obj.version;  
  ge('ubuVersName').value = obj.release;  
  
  var editButton = ge('submit');  
  editButton.value = 'Edit Version';  
  editButton.key = key;  
}
```

based on the key that was passed to the function, just like we did for each of the rows in localStorage when we were displaying them. Instead of displaying them, though, we are setting the value for the form fields to the information we get back.

After the fields are repopulated with the information, we set the submit button to display “Edit Version” instead of “Add”, and tie a key to its object with the value of the key we are editing. That way, when they press the submit button, it is doing the same thing it normally would, only this time it will trip our submit.key exists if statement in our storeData function and be set to the value of the key object in our submit button object. Pretty cool, eh?

Our last function is a very, very easy function. However, it has been elaborated and filled up with a lot of extra lines it doesn’t need, just to make sure the user actually wants to do it. It is the clear localStorage button function: <http://pastebin.com/i9shwb7z>

The only line we really need in this whole function is localStorage.clear(). However,

when this is run, it will clear everything in it, no going back. With that said, the first thing you are going to want to do, is make sure that there is anything in there in the first place and let the user know if there isn’t. If there is, and it will be removing data from it, ask the user if he/she really wants to do this. The importance here is that localStorage can contain information from your site, and from others. It isn’t like a cookie that can be accessed only by your domain name – localStorage can be read and manipulated by any site, any domain, any page. If they do indeed want to delete, remove it and call our displayData() function to display nothing in our list.

Now that we are done with all of our functions, it is time to tell the browser what to do when it loads our javascript: <http://pastebin.com/VqzFiDYF>

The first thing is a variable for our form. Next, we are attaching the processForm function to our form when a user submits it. To end our javascript goodness for the month, we are calling the displayData() function so that when a user comes to our page, it will pull up any information he

previously entered into our form when it loads.

I very much hope you enjoyed this month’s article. I am planning on showing you how to convert your code to JQuery, add a little bit of “zaz” to our page, and moving our localStorage to a real database. I would love to hear your suggestions for any future web developer articles you would like to see. Feel free to shoot over your suggestions or any other comments via twitter: [@aliendev2](https://twitter.com/aliendev2).

Remember to check back next month for some more web developer awesomeness. You can also check out, follow, fork, download, view revisions of the code on github: <https://github.com/aliendev/FCM-UbuntuVers>.



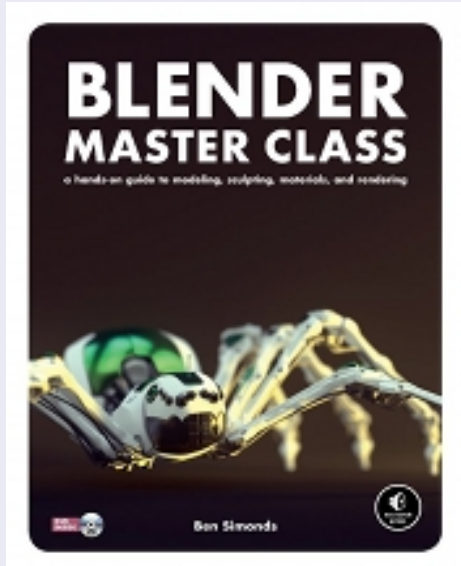
Michael Youngblood has been in the industry of web design and development for 13 years. He has been working for a world wide wireless tech corp for six years and is working on his bachelor’s of science in mobile development.



40% OFF BLENDER MASTER CLASS

For one week only, get 40% off Blender Master Class and learn the secrets of 3D design from a master artist.

Use coupon code **WILLITBLEND**



Coupon good on print or ebook purchases. Print books come with free ebook editions (DRM-free, of course).

http://nostar.ch/Blender_promo



HOW-TO

Written by Ronnie Tucker

Write For Full Circle Magazine

GUIDELINES

The single rule for an article is that **it must somehow be linked to Ubuntu or one of the many derivatives of Ubuntu (Kubuntu, Xubuntu, Lubuntu, etc).**

RULES

• There is no word limit for articles, but be advised that long articles may be split across several issues.

• For advice, please refer to the **Official Full Circle Style Guide:**
<http://url.fullcirclemagazine.org/75d471>

• Write your article in whichever software you choose, I would recommend LibreOffice, but most importantly - **PLEASE SPELL AND GRAMMAR CHECK IT!**

• In your article, please indicate where you would like a particular image to be placed by indicating the image name in a new paragraph or by embedding the image in the ODT (Open Office)

document.

• Images should be JPG, no wider than 800 pixels, and use low compression.

• Do not use tables or any type of **bold** or *italic* formatting.

If you are writing a review, please follow these guidelines :

When you are ready to submit your article please email it to:
articles@fullcirclemagazine.org

TRANSLATIONS

If you would like to translate Full Circle into your native language please send an email to ronnie@fullcirclemagazine.org and we will either put you in touch with an existing team, or give you access to the raw text to translate from. With a completed PDF, you will be able to upload your file to the main Full Circle site.

REVIEWS

GAMES/APPLICATIONS

When reviewing games/applications please state clearly:

- title of the game
- who makes the game
- is it free, or a paid download?
- where to get it from (give download/homepage URL)
- is it Linux native, or did you use Wine?
- your marks out of five
- a summary with positive and negative points

HARDWARE

When reviewing hardware please state clearly:

- make and model of the hardware
- what category would you put this hardware into?
- any glitches that you may have had while using the hardware?
- easy to get the hardware working in Linux?
- did you have to use Windows drivers?
- marks out of five
- a summary with positive and negative points

You don't need to be an expert to write an article - write about the games, applications and hardware that you use every day.





- ➔ Access all your data in one de-duplicated location
- ➔ Configurable multi-platform synchronization
- ➔ Preserve all historical versions & deleted files
- ➔ Share folders instantly in web ShareRooms w / RSS
- ➔ Retrieve files from any internet-connected device
- ➔ Comprehensive 'zero-knowledge' data encryption
- ➔ 2 GBs Free / \$10 per 100 GBs / Unlimited devices

<https://spideroak.com>

Online
BACKUP

Secure
SYNC

Easy
SHARING

Whether you need to access a document you have stored on a remote server, synchronize data between a Mac, Windows or Linux device, share important business documents with your clients, or just rest easy knowing all of your data is safely, securely, and automatically backed up - SpiderOak's free online backup, online sync and online sharing solution can handle all your needs!

SpiderOak offers a different approach to online backup by combining a suite of services into one consolidated tool - free online backup, synchronization, sharing, remote access, and storage. This difference is further measured in our zero-knowledge privacy policy - the first one ever employed in this setting. Our flexible design allows you to handle data from any operating system (Mac, Windows and Linux) or location (external drives, network volumes, USB keys, etc...) using just one centralized account.

Download mobile clients
for **iOS & Android**

JOIN SPIDEROAK NOW
Get 2 Free GBs

Get 25% off any SpiderOak package
with the code: **FullcirclemagFans**



ASK THE NEW GUY

Written by Copil Yáñez

Hi, everyone! Welcome back to Ask the New Guy!

If you have a simple question and want an answer that doesn't require taking sides on the "round vs. square dialogue box corners" debate, contact me at copil.yanez@gmail.com.

Today's question is:

Q: I want to watch cat videos. How do I do that with Ubuntu?

A: Good question. Nevermind all that tosh about a distributed communication system that could survive a nuclear war. The Internet was designed first and foremost to allow Department of Defense honchos to exchange videos of Mr. Tibbles attacking a potato.

Early cat videos were nothing more than ASCII animations that Richard Nixon screened for his dog, Checkers, but we've come a long way since then. Now we have a whole universe of high-def movies, TV shows, and dad-

getting-hit-in-the-crotch-at-a-kid's-birthday-party footage. Accessing those videos is the first stop for many of us once we get Ubuntu installed.

But, when you go looking for high-larious cat vids, you'll quickly discover that not ALL videos are available to you. What gives? This is not the America Nixon would have wanted!

Don't panic. All that feline tomfoolery is only a few clicks away. Let's look at how to get videos working in Ubuntu and where to go for easy, no-hassle streaming on-line.

First, though, a quick word on philosophy, because I know how much readers of this magazine like to get their Hobbes on.

Ubuntu, like Linux itself, is built on the open source ideal that software should be free to download, modify and distribute. Unfortunately, many of the videos we want to watch are encoded using proprietary software, known

as codecs. Codecs are easily available, and, once installed, allow you to watch just about any video format out there.

So, why aren't they available by default? Well, including them would do two things. First, it would require that all users pay a fee for the privilege. Second, it would undermine the very philosophy on which Ubuntu was built and which encourages the smart minds helping in its development to contribute.

Are you doing something wrong by installing the proprietary codecs?

Um. Yes? No? Maybe?

Here's where things get a little gray, especially in the US where use of these codecs is covered under the Digital Millenium Copyblah, blah, blah, and I've lost you. Look, even the most ethical examination of the issue would still argue for the personal use of proprietary codecs since you still have to pay for content that's been

encoded using them. And, while piracy is certainly a concern online, all the examples you'll see below are legal streaming services.

Besides, the consequence for not using the codecs is that you'll have to get your cat videos the old fashioned way, via zoetrope and flip books. At that point, you're just one pair of aviator goggles and a walking cane away from becoming a character in a bad steampunk novel.

So what codecs should you get and what will they allow you to watch and how do they work?

Who cares?

Remember, this is Ask the New Guy. My goal is to get you up and running without too much mucking about. Let's just download the most common stuff and move along.

In Ubuntu 12.10, click on the Ubuntu Software Center icon along the left side of your screen. If it's not there, you can also get to it by



typing “soft” into the Dash search field.

When the Ubuntu Software Center screen opens, type “restricted extras” in the search field. You’ll see some software called “Ubuntu Restricted Extras” (funny how that works, no?). Click on that and then click Install. You’ll have to give Ubuntu your password to finish the install.

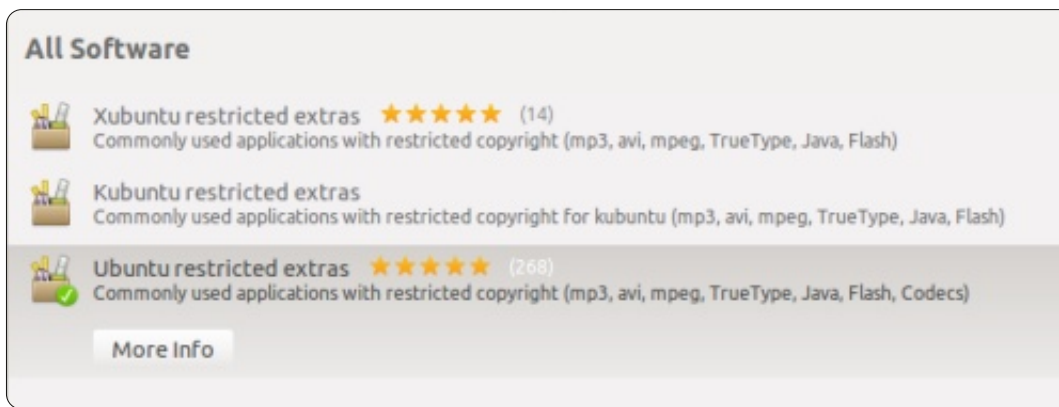
Because some of the fonts that are included are owned by Microsoft, you’ll need to click on their EULA (End User Licensing Agreement). I didn’t read it all but you’re welcome to. Pretty sure I owe them my spleen now.

Once you’re done with that, you’ll probably also want the ability to play DVDs by typing the following into a terminal window (CTRL-ALT-T):

```
sudo
/usr/share/doc/libdvdread4/in
stall-css.sh
```

Um, that’s kinda it!

I know, everything in Ubuntu is so easy, it’s anti-climactic. It’s like putting together a crack team of



operatives to steal the Crown Jewels – only to have the Queen herself hand them to you while she’s out for a pint at the corner pub.

Yes, you may run across some formats that won’t play automatically. The error message usually provides a link to search for and download the appropriate codec. If that appears, do it, for God’s sake! Cats playing pianos, man!

So, you’re set up to watch all manner of videos in formats you didn’t even know existed. What’s out there? May I suggest you head over to YouTube and search for funny cats? We’ll see you in a few weeks.

While you’re there, did you know you can also rent movies

from YouTube? That’s right. Click on <http://www.youtube.com/user/movies> and watch The ABCs of Death or All Superheroes Must Die. Oh, I forgot, you’re so highbrow, la-dee-da, too good for The Human Centipede. Fine. They’ve also got Wall-E. Happy now?

If you have Amazon Prime, you also have a ready supply of FREE streaming movies, including Mission Impossible III and Downton Abbey. If you have problems with the videos not playing, run these commands to allow the Amazon player to update:

```
sudo apt-get install hal
cd ~/.adobe/Flash_Player
rm -rf NativeCache AssetCache
APSPprivateData2
```

Hulu Desktop has a version that works with Ubuntu, and you can download that here <http://www.hulu.com/labs/hulu-desktop-linux>. If you’ve been looking to catch up on the adorkable Zooey Deschanel on New Girl, now’s your chance.

Crackle.com is a relative newcomer and offers no-hassle streaming of movies and TV shows. It’s not the most current selection, and you do have to sit through commercials, but c’mon, three words: Beverly Hills Ninja.

One last suggestion for you. Check out www.canistream.it. Enter a movie you want to watch and the website will tell you which service, if any, streams it.

Well, I guess that covers it for – what’s that? Netflix? Um, yeah, they’ve been promising native Linux support for a long time now. Thankfully the community has come to the rescue. There’s a PPA which can be added to your list of repositories that will let you watch Netflix in Ubuntu. Run these two commands in a terminal:

```
sudo apt-add-repository
ppa:ehooover/compholio
```

```
sudo apt-get update && sudo
apt-get install netflix-
desktop
```

Voila! You'll now see 'Netflix Desktop' in your list of applications.

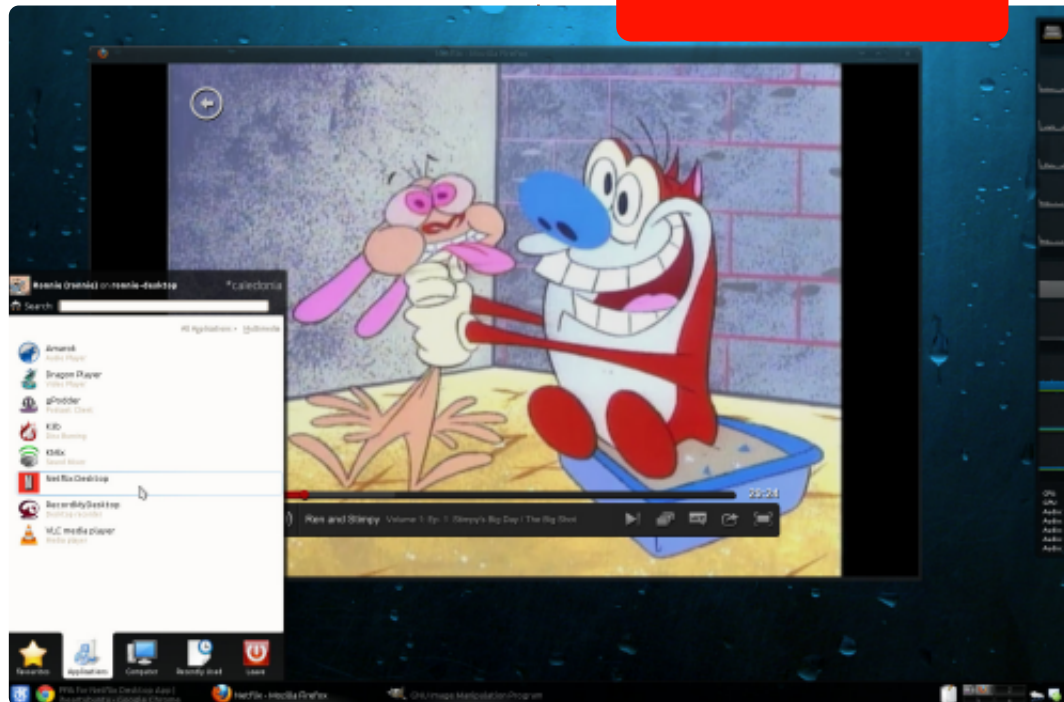
Well, hopefully you had no problem setting up video playback in Ubuntu and found something worth watching. Any problems? Let me know.

Good luck and happy Ubuntuing!



Copil is an Aztec name that roughly translates to "you need my heart for what again?" His love of women's shoes is chronicled at yaconfidential.blogspot.com. You can also watch him embarrass himself on Twitter (@copil).

NETFLIX



16x16 SUDOKU

Numbers 0 to 9 and letters A to F are to be filled into the 16x16 grid so that every row, every column, and every 4x4 box contains 0 to 9 and A - F.

E		D	9		2		0							5	
3		4				5			C		A	F	6	1	9
	6	C	5			9	1		8				E		
1		B			E	6	C		D	5		4			2
			3			4	6				C	7	1	A	
				2	0	3			9		1		5	D	F
6			0	E						B			C	9	
	D	5	B	1				A			2				
				9			A				0	3	4	F	
	3	E			5						8	B			C
B	2	9		0		E			A	1	4				
	A	1	C	6				D	5			0			
2			E		9	F		0	1	C			B		6
		3				D		F	B			1	0	E	
8	9	0	1	C		B			2				F		4
	B							7		4		9	2		5

Solutions are on the second last page.

Puzzles are copyright, and kindly provided by,
The Puzzle Club - www.thepuzzleclub.com



In FCM#61, I wrote about DVD ripping and encoding. For the article, I tested the ripping and encoding speeds on single core, dual core, and four core systems. As expected, the four core systems had the fastest speed from start to finish. But there was a surprising result as well, the DVD burners in the dual core systems ripped (cached the DVD to the hard drive) almost 3 times faster than the quad core systems (and just less than double an eight core system armed with a SATA Blu-ray I tested later). From this, I surmised that the performance of the DVD players/writers had more to do with the caching speed of the movie than the speed or number of CPU cores.

Why does all this matter? DVD caching (commonly called ripping) takes only a fraction of the time that's needed for compressing and encoding the video, but it still adds several minutes to the process. Ideally, you want the best outcome on the best possible hardware. If you have 100 DVDs to back up (caching is used in backing up

DVDs as well), and one DVD drive takes 10 minutes longer, that process is suddenly 1000 minutes (16.67 hours) longer.

Logically, you would think the latest technology would give you the best results. But what about across brands?

To compare DVD ROMs and writers, I used a selection of DVD drives from different years, using

different interfaces, and from different manufacturers. I used Acidrip to cache the DVD. Each time I cached a DVD, I made sure to use a new directory for the cache (deleting any old directories to ensure Acidrip wasn't borrowing from any other existing cache).

The results are shown below.

In this case, the newest technology didn't necessarily mean

the fastest. Although I didn't test a Blu-ray player for this experiment, I did test one prior to this experiment using the same DVD I used in the experiment in issue 61, and (on the 8 core) it cached better than most, but still a couple of minutes slower than the older DVD writers in the dual core systems, just over 6 minutes.

Interface also doesn't seem to make a big difference. Though I tested only 1 SATA DVD drive (and the SATA Blu-ray I mentioned), a couple of PATA DVD drives beat the SATAs.

Both the slow Hitachi-LG and Sony DVD drives had small buffer sizes. You could surmise that the buffer size has something to do with the speed of the caching, except for the fact that the fastest AOpen DVD drive had a smaller buffer than most of the other drives.

Made in 2001, the AOpen DVD drive is also one of the oldest drives, but it posted the best caching time, so age isn't

DVD Mfg & Model	Mfg Date	Interface	Type	Min. to Cache	Max DVD read speed	Buffer	DVD avg access time
Hitachi-LG DC-4522B	April 2005	PATA	DVD-R/CD RW	10.09	16x	2MB	120ms
Hitachi-LG GDR-H30N	July 2007	PATA	DVD-R	14.46	16x	256KB	100ms
Lite-ON SHW-1635S	September 2005	PATA	DVD-RW/CD DRW	4.33	16x	2MB	160ms
Sony DDU220E	February 1999	PATA	DVD-R	14.11	5x	512KB	115ms
TOP-G BDV 212B	September 2001	PATA	DVD-R	6.4	12x	256KB	110ms
Toshiba Samsung TS-H493	January 2008	SATA	DVD-R/CD RW	8.55	16x	2MB	150ms
AOpen DVD1648	November 2001	PATA	DVD-R	4.32	16x	512KB	110ms

necessarily a factor (though wear and tear could be).

What about the Maximum DVD read speed? The Sony DVD drive was the slowest in the group and the second slowest performing drive (not to mention the oldest), but it still beat a drive eight years newer and with the fastest Maximum DVD read speed.

One rumour I heard was that certain manufacturers purposefully cripple their drives to make it more difficult to copy DVDs. Looking at the statistics, the worst performers are Hitachi-LG drives, and the Sony drive. While Sony is a well known supporter of Digital Rights Management (DRM), and LG drives were among the slowest in the test conducted in FCM#61, I'm skeptical this is the reason why these particular drives performed the worst.

Doing more digging, I discovered that the maximum DVD read speeds published by manufacturers are not necessarily the maximum DVD video read speeds. This is where it gets tricky; it's difficult to find accurate DVD video read speeds for drives.

I found that the LG DC-4522B, rated for 16x, actually transfers video data at a maximum of 4.8x. The maximum DVD read speed is also part of a group of statistics known as Data Transfer Rate (DTR) which includes burning as well. Drive manufacturers either no longer have – or have never -- published DTR video statistics for the other drives in this experiment.

At this point, I went looking to other sources and found www.cdrinfo.com – a site (with banner ads) that publishes a variety of information about

optical media technology. The site is Windows-oriented, relying on proprietary programs to do a lot of the testing, but their numbers seem to jive with the data I gathered. The AOpen DVD1648 was among the fastest of the DVD drives they tested copying DVDs using CSS. While cdrinfo.com didn't have the same LG drives tested here, the LG 16x drives I looked at were among the slowest drives reading CSS encrypted DVDs.

So what does all this mean? If you're interested in archiving your DVDs or Blu-rays, try to get as

much information about the DVD video read speeds beyond the simple maximum read and burn speeds published on the websites of most manufacturers. A speedy DVD or Blu-ray drive could save you hours of work.

IMPORTANT LINKS

<http://www.cdrinfo.com/>
<http://www.btc.com.tw/>
<http://www.lg.com/>
<http://global.m.aopen.com/>



Charles is a step-father, husband, and Linux fan, who hosts a not-for-profit computer refurbishing project. When not breaking hardware/servers, he maintains a blog at:

<http://www.charlesmccolm.com/>



MY STORY

Written by Robert Boudreau

I should preface this by saying that I've been using Linux for a long, long time. I also should note that, in all that time, I've never learned to program, develop or hack anything. I can write basic scripts if I have to, tinker with config files when necessary, and work when needed with the command line; that's about it. I'm a user, not a developer, hacker or guru.

I first came to Linux in 1995, after repeated headaches with the then new Windows 95. I'd heard about this Unix-like operating system that a college student had created and was giving away for free, and it piqued my curiosity. I'd like to say I just found a distro and installed it and life was grand, but it didn't work that way back then. I instead had to download the source code, get a GCC compiler and all the other necessities running under DOS, and do it the hard way. But I was hooked. I started with TWM as a window manager, compiled a few apps I downloaded from a BBS, and I was off and running. Within a year or

so, I'd found you could buy packaged versions at the local office supply, and my first was Red Hat, though now I can't remember what version it was. Later came Debian, Mandrake, then SuSE. Debian and SuSE – now openSUSE – I've bounced back and forth with over the years, and for perhaps the last five or so I've pretty much settled on openSUSE. Until just recently.

Anyone who keeps up with Linux news in recent years can't avoid running across things about Ubuntu. The trolls may say what they will, but Ubuntu has made great progress in putting Linux into the mainstream, and today you can't find many applications that, if they have a Linux version at all, aren't packaged for Ubuntu. Many computer manufacturers, like System76, if they use Linux, it's

Ubuntu. It's in schools, institutions, government offices, it's just about everywhere. Even Google now has its own version of Ubuntu for its employees. What is so special about it? Why has one distribution become almost synonymous with Linux? It piqued my curiosity.



About two years ago, I actually tried Ubuntu – I think it was version 10.04 or 10.10, and frankly wasn't that impressed. It was Gnome-centric

for one, and I'm a KDE user. I read about Kubuntu being a KDE spin of Ubuntu, so I gave that a try. Sorry to say, compared to openSUSE or Debian it was too slow and bloated for my taste, so it too didn't last long on my computer. In the time between then and now, though, Ubuntu has continued to become the port of first call for anyone looking into or

trying Linux. Even one of my favorite video podcasts, the Linux Action Show, has become very Ubuntu-centric, Ubuntu gaining converts of the two hosts. So, recently, I decided to use the example of one of the hosts and give the latest Ubuntu a 30-day tryout, basically force myself to use it for 30 days and see how it felt after that. About two months ago I started on this experiment, and, well, here's what happened.

At first, I was quite impressed with Unity, it was different than the Gnome 2 I remembered, but more impressive to me were the speed improvements since my last foray with Ubuntu two years ago. However, after just a week of using only Unity, I started to feel that, while I'd learned my way around it and most of its way of doing things, I saw it as more eye-candy than as a better way of working. While I liked some of the lenses and the way they integrated the web into the desktop, I felt overall the HUD was too slow, took up too much screen real estate, and, in particular with the menu

integration, it was just too hard to find what you're looking for. I know it's a work in progress, in time it will get much better and more refined, but, as it stands now, it just doesn't work for me and the way I like to do things. But, I was going to be fair, and stuck to my plan of using it for thirty days. Then I dumped it and installed openSUSE 12.1.

Something happened, though. There was an unexpected side-effect. I'd never even given it a thought while I was experimenting with Ubuntu, but it was there nonetheless, and, when I went back to openSUSE, it suddenly dawned on me. I knew why Ubuntu and its spin-offs are always at the top of the distro heap, why they're so popular, and why they're achieving what so many others have failed to: it all has to do with packages.

For those unfamiliar with openSUSE, it's a great Linux distribution, one of the best. They have huge software repositories with just about anything you could want in them. But they have one serious drawback: you have to be real careful if you install software from any but the default

repositories. If you do want to get a newer version of something, or to try some software from another repository, it's often going to lead to dependency problems. Then there's the problem of trying to find and install proprietary drivers or software that doesn't fit the "100% Open Source" model, like the VLC media player. OpenSUSE has a great software search system on its web site, and you can almost surely find what you want, but it requires adding another repository that in all likelihood will cause conflicts.

I, of course, had learned to deal with these problems years ago, and could generally work around them

to get the system I wanted, but it's still a bit annoying. And, very often when it came time for a program upgrade, I was left having to repeat the whole dependency fixing all over again. After I did this last install of 12.1, and went through all the fixes that I needed to do in order to get my computer where I wanted it, it hit me. I didn't remember having any need to do that in Ubuntu. Even when I added and pulled software from PPAs, there weren't any conflicts. No dependencies to hunt down, nothing that worked before and then got broken. It all just worked.

That's Ubuntu's greatest advantage, why I believe it has

gotten to be the default choice of not only new Linux users, but also many seasoned veterans. The Ubuntu packaging system is second to none. I doubt this is easy to do, but it seems to be a core duty of the packagers and those who oversee the repositories, that things are gotten right. It's obviously why so many Ubuntu variants and spin-offs still use the Ubuntu repositories, they can count on that stability. In my opinion, it's a major achievement what will keep Ubuntu in the forefront. Not the snazzy desktops, the coming web integration, or the constant pro and con media hype that surrounds it. It'll be that rock-solid packaging system that will keep Ubuntu at the top.

When I realized this, what did I do? Installed Kubuntu. I did mention I was a KDE guy, didn't I? What a difference from just two years ago! The latest Kubuntu (12.04.1) is as solid, fast and well-done as any KDE distro out there, and... it has those great Ubuntu repositories behind it. What more can anyone ask for?



MORE UBUNTU!

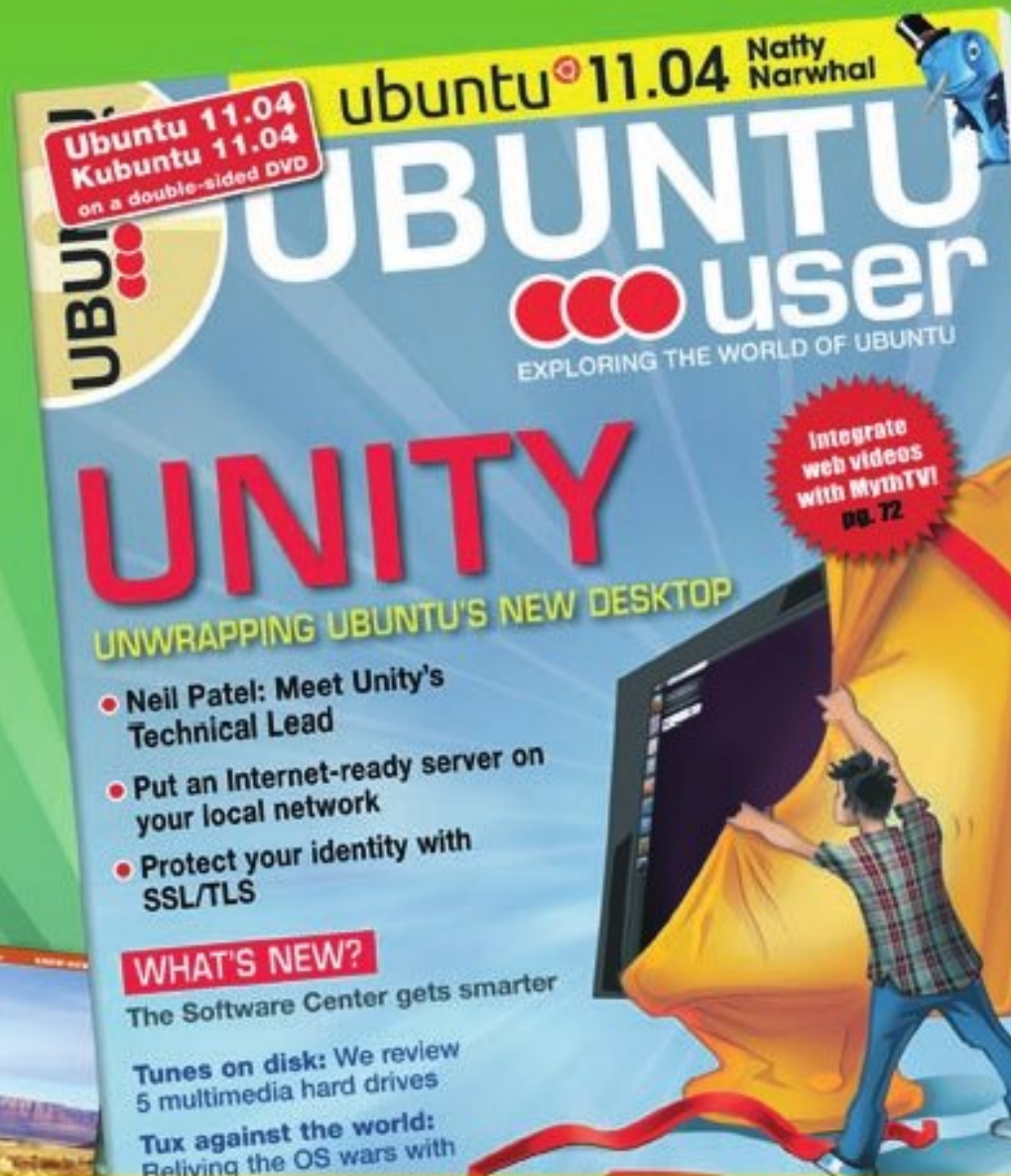
Can't get enough Ubuntu?
We've got a whole lot more!

Ubuntu User is your roadmap to the Ubuntu community. In the pages of **Ubuntu User**, you'll learn about the latest tools, best tricks, and newest developments in the Ubuntu story.

DON'T MISS ANOTHER ISSUE!



UBUNTU-USER.COM/SUBSCRIBE-NOW



FOLLOW US ON

TWITTER: UBUNTUSER

FACEBOOK: UBUNTUSERMAG



REVIEW

Written by Lucas Westermann

HTC One S



There has been quite a buzz surrounding the HTC One X – reviews, questions about the release of Android 4.1, etc. However, it's rather uncommon to hear much about the One X's little brother – the One S. The HTC One S is essentially a smaller, cheaper, and slightly less powerful version of the One X. Having had (and used) the phone for a good few months, I felt it time to share my insights into the device. The phone I was using before this was a Samsung Galaxy S II, which is what I will be comparing the One S to.

Specifications

- 130.9 x 65 x 7.8 mm (length x width x depth)
 - 120 grams
 - 4.3" AMOLED display with a 960 x 540 (qHD) resolution
 - Android 4.1.1 with HTC Sense 4 (may still be shipping with 4.0 – update is available over the air)
- Complete list:

<http://www.htc.com/www/smartphones/htc-one-s/#specs>

SCREEN

At first glance, the screen seems a little less vivid (compared to the S II). Though the colors aren't quite as bright, I find the HTC One S to be a lot more comfortable on the eyes, and the screen seems much more responsive. The real estate is used well by the Sense skin, and I haven't noticed any contrast issues

with the official themes for Sense. The automatic backlight adjustment is adequate – I've found that at times the screen can be a little dark for my tastes, which is easily remedied by adjusting the brightness manually. The only issue I have is the "ring toss" slide to unlock feature. It's fine if you're planning to just unlock your phone – but it's also the method needed to answer calls when the phone is locked. If you're not careful enough to get the "answer" icon into the ring, it will fail to answer the call, and you'll have to try again. Very problematic if you're on the go and have cold hands (or are wearing gloves which you need to take off first). More than one call was nearly missed this way.

CAMERA

HTC advertises the One S as a

great photo-taking phone, offering things such as panorama shots and low-light capabilities. Personally, I find the camera works better than any of my past cell phones – but it doesn't beat most point and shoots. Features such as the panorama mode can be buggy (or just plain difficult to use). If you didn't own any other camera, you could probably manage to do very well for yourself if you invested in some form of tripod to keep the camera steady. Also very usable if you're out somewhere and want to capture a moment. Reason alone to buy the phone? No. If you're torn between two phones and camera quality is important to you, then it might help make your decision.

As you can tell, the photos are nice, although the first shot of the panorama (I went right to left) is



extremely blurry, though I felt that I was holding the camera steadiest for that shot. If anyone is interested in seeing the original photos, send me an email and I'll pass them along.



PERFORMANCE

The device has a 1.5GHz-1.7GHz dual core CPU, which results in a

snappy response from the device at most times. Opening apps (besides games, which take just a little bit longer to open than normal apps) is quick, and using the Recent Apps feature is a dead simple way to free up some of the 1GB of RAM, if you need it. Wi-Fi signal strength is on-par with any of my laptops, or my Asus TF101 tablet. This means that I have a steady (if sometimes weak) connection, even in some of the problem rooms where the Galaxy S II had issues. I also find that the One S is faster to change from E to 3G connection when it becomes available – the Galaxy S II always had a delay before the speed adjusted, which doesn't seem to be present in the One S. Definitely an improvement if you're someone who commutes through many areas of varying signal strength. That being said, the phone does seem to ignore areas of extremely weak signals entirely. The S II often had a weak connection in places where the One S has none at all. However, this won't be typical for anyone who lives in an area with even sub-par coverage.

The One S responds to touch input very well, with the exception of the lock screen, which seems to

be more of an implementation issue than an actual hardware issue. I've also noticed a significant improvement on transitions and web browsing with the upgrade from 4.0 to 4.1.1 (and the addition of Project Butter). The difference between the One S and the Galaxy S II is the difference between night and day. The S II could freeze or lag when getting around the home screens, whereas the One S handles it without so much as a hiccup. There are, obviously, some performance issues that crop up when you're taxing the device with heavy apps – but for normal basic usage, it's a vast improvement. However, the HTC One S does not contain a removable battery – so if it does hang, you'll need to hold the power button for 10 seconds and cause a “hard reset”, which is useful to note. Also, I've found that this can sometimes result in your phone rebooting in your pants pocket, as the power button is located directly at the top of the device. If you keep your devices in a jacket pocket, shirt pocket, or a purse, you probably won't notice this issue. That being said – the One S has also randomly rebooted while I was using it. It happens very rarely, but not for any reason I can discern. This could also be

happening in my pocket, rendering that complaint moot. Is a reboot problematic? Not really, as the device boots quickly, and I have yet to experience a crash when doing things like talking on the phone or replying to emails/text messages. The S II would hang with surprising frequency when doing any task at all, leading to a few interrupted calls.

SOUND QUALITY

HTC loves to advertise that the One S comes with “Beats Audio”. However, I don't find that the audio quality sounds any better/worse than my Motorola Milestone, the Samsung Galaxy S II, or my old 2nd generation iPod Touch. Not that I have any complaints about the quality. The speaker is sufficient (though the S II seemed to be a bit clearer), and phone calls are extremely clear. I have yet to have complaints about the call quality from the device (even in crowded subway trains). As such, I'm inclined to say the microphone (and noise filtering) is a lot better than the S II – though I can't say I've done a fair trial.

BATTERY LIFE

I haven't put the phone through a stress test of any sort to measure exact hours under heavy load. However, I tend to use my phone for listening to music whenever I go out, it's set to automatically pull/push emails, and if I am using the maps function, I'll have Wi-Fi running in the background. Otherwise I shut off all unnecessary antennas (Bluetooth, Wi-Fi, etc). Throughout the day, I'll answer emails, check news, browse websites, download some updates, upload photos via instant upload on Google+, use Dropbox, occasionally tether the device via bluetooth, or use it for navigation. At the end of the day, after about 6-8 hours of usage like that (with constant music playback) the device is around 35-40%. Occasionally it drops into the red 14-15% mark, but I have yet to experience the phone being dead when I need it. I do charge it every night (unless it's nearly full and I'm at home the next day so using the phone is unimportant, then I may just leave it). I also put it into airplane mode when I'm at home, as my house has terrible reception – I save some battery life by avoiding the search for radio towers. If I need to find a signal, I can quickly turn on the antennas

again. Overall, the device seems to last longer than the S II, and is roughly on-par with my father's iPhone 4S.

BUILD QUALITY

The phone seems to be made out of metal, for the most part. The screen also seems to be made out of tough glass – I can't say for certain what exactly it's made of, but it resists any scratches from whatever I may have in my pockets (coins, a zipper from my headphone case, etc). It does collect a lot of fingerprints (obviously) but they wipe off without a lot of effort – the screen has probably been treated somehow. Overall, the device feels solid and has survived a few roughly one-meter falls without a scratch (I should probably note that these were neither intentional, nor caused by me). The

covering for the SIM card is solid, and can probably survive being taken on and off a great number of times. The camera stands out a little at the back, where I would have expected it to collect a lot of scratches. However, the lens seems to be made of the same material as the screen, and has yet to get a serious scratch.

CONCLUSION

The HTC One S is a great little phone that can currently be had for around €380. The Samsung Galaxy S II I owned cost about €20 more, and I was never half as satisfied with it as I am with the One S. If anyone is in the market for an Android smart phone (that isn't quite as huge as the One X), then I would definitely recommend that you consider the One S when making your decision. If you want a phone that can double as a camera

on the fly, the One S is by far the most capable camera replacement I've had the pleasure to use, so also consider that. For anyone who is looking for a larger device, the One X should definitely also be on your list of options, as its quality is very similar to the One S.

A generally great device, though the occasional random crash, and the somewhat inaccurate “slide to answer” system prevents it from getting full marks.

If anyone has exact questions, or wants to know something I haven't covered, feel free to email me at lswest34@gmail.com. Please put “HTC One S Review” or “FCM” in the subject line, so it doesn't disappear into my inbox.



Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.





LETTERS

If you would like to submit a letter for publication, compliment or complaint, please email it to: letters@fullcirclemagazine.org. PLEASE NOTE: some letters may be edited for space.



Join us on:



goo.gl/FRTMI



facebook.com/fullcirclemagazine



twitter.com/#!/fullcirclemag



linkedin.com/company/full-circle-magazine



ubuntuforums.org/forumdisplay.php?f=270

THINGS ARE HEATING UP

In FCM#69, reader Saravanan M observes that his Dell Vostro 1550 laptop runs too hot under Ubuntu, but not under Windows 7. I have seen similar things happen to a variety of laptops – though not all. The only common points with computers that exhibit this behaviour are a 64-bit CPU and a dedicated graphics chip, i.e. besides the one integrated into the CPU. For example, it has happened to me with a Lenovo (Core i5 and Nvidia graphics card) and a Sony Vaio (Celeron with a secondary Intel GPU), both at least since version 12.04 onwards. It also seems to be the case for the Dell that Saravanan mentioned.

A solution that worked for me was to abandon Ubuntu and derivatives for these computers, install Debian (or Linux Mint Debian) instead. The fans no longer spin up constantly, and battery life got rather better.

I am not very clear as to why this happens, but the fact that

Debian tends to be slightly more conservative towards introducing modifications into the kernel may be a factor. The following Launchpad thread gave me food for thought:

<https://answers.launchpad.net/ubuntu/+source/linux/+question/195794>

Alan Ward

AMAH! HOME SERVER

Danish Lala asked about an article for creating a Home Server running Ubuntu. I'd like to point out that **Amahi Home Server** is setup with either Fedora or Ubuntu 12.04 as the operating system. It does almost all of the heavy lifting for you. It can be found at: <http://www.amahi.org>.

Patrick Dickey



Amahi

Full Circle Podcast Episode 32, The Year That Was....Well Nearly!!

Your hosts:

- Les Pounder
- Tony Hughes
- Jon Chamberlain
- Oliver Clark



from the Blackpool (UK) LUG
<http://blackpool.lug.org.uk>

In this episode Les brings everyone up-to-date with where he's been, there's a "Special Announcement" and the lads take a look back at 2012, "A Year in Tech"

Download

FULL CIRCLE NEEDS You!



Without reader input **Full Circle** would be an empty PDF file (which I don't think many people would find particularly interesting). We are always looking for articles, reviews, anything! Even small things like letters and desktop screens help fill the magazine.

See [page 29](#) to read our basic guidelines. Follow those and you're almost guaranteed success.

Have a look at the last page (of any issue) to get the details of where to send your contributions.





Q&A

Compiled by Gord Campbell

If you have Ubuntu-related questions, email them to: questions@fullcirclemagazine.org, and Gord will answer them in a future issue. Please include as much information as you can about your problem.

Q I carelessly deleted the /tmp directory. Now I can't login.

A (Thanks again to **SeijiSensei** in the Ubuntu Forums) See this message thread: <http://ubuntuforums.org/showthread.php?t=2107154>

Q I have a Macbook Pro with Retina display. Can I use Ubuntu?

A You're going to love 13.04! For now, see this blog: <http://linuxmacbookproretina.blogspot.com.au/2012/12/ubuntu-1304-daily-build-macbook-pro.html>

Q After installing Ubuntu 12.10, Libreoffice won't let me open any bases or build new bases. The message says it can't locate any Java installations. Has anyone else had this problem?

A Yes, I would say the the packaging of Libreoffice-base is incomplete. After installing **openjdk-jre**, you should be able to

create a database.

Q How can I make my wireless connection use a static IP address in Ubuntu 12.10?

A (Based on information from **chili555** in the Ubuntu Forums) This is a blow-by-blow, which should also work in Ubuntu 12.04, and you can probably do it faster than the time it takes to read this. Click on the network icon, which is probably next to the volume control on the top-right of your screen, and select Edit Connections. Select the Wireless tab. Click on the name of your current connection and select "Edit".

A new window pops up. Select the IPv4 Settings tab. "Method" is a drop-down box, select "Manual." The next box is Addresses, click on Add. For "Address," type in the address you want the computer to use. Make sure it is not in the range of DHCP addresses your router might assign. I used 192.168.1.32 For Netmask, use

255.255.255.0. Gateway is the address of your router. Mine is 192.168.1.1 but some routers use 192.168.0.1. Some use a very different address.

In order to browse the web, you must specify DNS Servers. When you signed up with your ISP, you probably got an information package suggesting what DNS servers to use. Google operates excellent DNS servers at 8.8.8.8 and your router can also pass requests to DNS servers, so you might enter 8.8.8.8 192.168.1.1 into this field. I plugged in the numbers my ISP suggested, separated by a space.

Save your changes, and you're done. You can restart networking, or just reboot to have the change take effect.

At least, that's the theory. I discovered that Network Manager saved the original settings under a new name, and that is what it used. When I clicked on the icon, it didn't offer the modified settings as a connection option. So, I made the

changes once again, this time to the new name. Clicked save, disconnected and reconnected, and I was using the static IP address. I used Conky to display my IP address with this line:

```
IP Address: ${addr wlan1}
```

Your wireless might be wlan0.

Q After entering all the necessary info to set up the wireless connection, not only did it not connect, but there did not appear to be any attempt to connect. Went to the settings/network/settings page and it reports that it's on, the hardware address, and also says "firmware missing".

A (Thanks to **ubfan1** in the Ubuntu Forums) From a wired connection, add the package linux-firmware-nonfree

```
sudo apt-get install linux-firmware-nonfree
```



Q After installing Ubuntu 12.04, my G4 Mac mini was mute.

A (Thanks to *linuxopjemac* in the Ubuntu Forums) Use this command:

```
sudo nano /etc/modules
```

Add these lines at the end:

```
snd_aoa_i2sbus
snd_aoa_fabric_layout
snd_aoa_codec_tas
snd_aoa_codec_onyx
```

Then CTRL-X and "y" to save. Reboot. Sound works, hopefully.

Q (Thanks to *baseballa51* and *DuckHook* in the Ubuntu Forums) I have a RocketRAID 2640 x1 PCIE 1x 4 port SATA Controller, with four drives attached to it. I cannot access my HDDs connected to the RAID controller card.

A This web page: <http://www.flynsarmy.com/2012/11/installing-rocketraid-2760a-drivers-on-ubuntu-12-10> was for a slightly different version of the card, but it spelled out the steps needed to use the RAID controller.

TIPS AND TECHNIQUES



Driver File Types

The Linux kernel is constantly growing, so new versions often drop support for older hardware. If you are stuck with one of these older devices, the solution can require compiling a driver from source code and installing it on your system.

That sounds drastic and complicated, but it's better than the alternative. When I moved from Windows 98 to XP, my lovely (and semi-expensive) webcam became electronic trash.

As *DuckHook* explained in this thread: <http://ubuntuforums.org/showthread.php?t=2101512> the first things you need to know about are some new file types. If you go to a vendor site and download a Linux driver for your device, you will probably get some-long-name-with-version-info.gz where gz is short for GNU zip. If you're coming from Windows, you are probably familiar with zip files, which may contain one or several files. The gz, or the zip, is interesting only once

you see what is inside.

Double-clicking on a .gz will generally open it in a program which can extract the contents. With any luck, there will be one or more .sh files, perhaps including install.sh

When you first look inside a .gz, you might see one or more files with readme in the name. Before you even extract the contents, you should see what they say. Sometimes they will bore you with a list of changes from the previous version, but sometimes they will contain information you really need to know. Then, you can extract the files.

A .sh is a shell script, similar to a .bat (batch) file in MS-DOS or Windows, but more powerful. It will probably compile the source code, but a default Ubuntu installation needs some enhancements to do this. The most basic is build-essential, so you should install that now. You might also need ubuntu-dev-tools; it never hurts to be prepared by installing that collection of programs. Once those are ready, you can run the script with these terminal commands:

```
cd the-folder-where-you-extracted-the-files
```

```
sudo sh install.sh
```

The gz probably contained some source code, but you don't need to know the file type, because the script looks after it. However, it might be useful to know what it produces: long-name-of-some-device-driver.ko where ko stands for kernel object. If you ever see instructions for installing a device driver, they may tell you to put a .ko file in a specific location.

The terminal command: lsmod will tell you what device drivers are in use. The results are normally short names which do not include version information.

DuckHook also explained that these commands:

```
sudo modprobe -r
name_of_driver
```

```
sudo modprobe name_of_driver
```

will unload and reload a device driver.

And always, remember: when you run across stuff you don't understand, Google is your friend.



Artizens is an upcoming co-op platformer with a focus on customization and creativity. Typically, a gamer is limited to the equipment presets, and high-level characters often look nearly identical. The team at Artizens, Inc. wanted to change things up a bit, and give players the chance to create their own weapons and armor.

Players have workshops where they create gear for their characters by uploading photos or drawing them themselves. For those not as confident with their

artistic skills, you can choose from various presets and gear created by other players from around the world. Once you decide on your equipment, you can mix them however you want, and share them with other players. If you're not keen on sharing your personalizations with the community, there will be a feature to make your character's gear invisible to everyone except for you. There won't be moderation on new uploads, but there will be options for flagging inappropriate or fraudulent items.

Artizens isn't just about customization though - the developers also wanted to create a different style of gameplay inspired by games like Monster Hunter and Magic: The Gathering. In each mission, players and their companions are called to battle through numerous arenas in pursuit of the stage's boss. Your skill rank will increase with each mission you complete, but the difficulty of the next mission will increase as well. If you lose a mission, your rank will go down and the missions will get easier. To keep things fresh and challenging, the missions are semi-randomly generated, and playing on a harder difficulty level will yield greater rewards.

Purchasing the game will give you 10,000 shines, the in-game currency. With the currency, players can buy additional content at the trading post, as well as in-game expansions. Shines can also be purchased with real currency, or earned by selling your content to other players. For those not interested in selling items, there

will be an à la carte system for acquiring new monsters, missions, and crafts.

Artizens, Inc. plans on releasing monthly expansions in the form of new monsters, crafts and mods. In addition to co-op, there are plans for a friends system as well as an online matchmaking feature. The team also plans to release frequent updates to improve player experience and to continuously add new content.

As of early February, Artizens has nearly reached its Kickstarter goal, and the alpha release is anticipated for July 2013. Artizens will be available for Windows, Mac and Linux, and the team hopes that there will also be future support for Steam.



Jennifer is a fine arts student from the Chicagoland area. You can follow @missjendie on Twitter or visit her blog at missjendie.com.



MY DESKTOP

Your chance to show the world your desktop or PC. Email your screenshots and photos to: misc@fullcirclemagazine.org and include a brief paragraph about your desktop, your PC's specs and any other interesting tidbits about your setup.



This is my Desktop. I love Linux and open source software.

My distribution is Ubuntu 11.10, and I use Conky and AWN on my desktop, and this is Gnome3 desktop environment.

CPU : Intel Core 2 Duo, 2.200 GHz
Graphic : Intel
RAM : 2 GB
HDD: 250 GB

Shell Theme : Dark Shine
Window Theme : Hope
Icon-Theme : Faience-Ocre
GTK+ Theme : Ambiance

Kaveh Shahhosseini



I run Ubuntu 12.10 Quantal with Unity 3D desktop on my Toshiba Satellite C655D-S5200 laptop with C50 Dual core 1.0 MHz processor and 4 GB DDR3 RAM.

The Wallpaper is Dark_Wallpaper_223 from <http://www.gothicwallpapers.com>, but I'm running Wallch, and changing them to the 1000 Wallpapers downloaded from the Wallch Website at 5 Min intervals.

I have a AMD Radeon HD 6320 Graphics chip-set so my drivers are not supported for Unity 3D. The Theme is Boje-Red running Compiz for Wobbly windows and Rain working. I'm saving for an Asus G74 so I can have the ultimate Ubuntu experience.

James Marshall





This desktop configuration runs perfectly both at my home and office workstations. I use Unity 5.16.0 together with Conky which is visible in the bottom right corner. I also use weather and system load indicators that can be seen in the top panel.

OS: Ubuntu 12.04.1 LTS i686
CPU: Intel(R)Core(TM) i5-2520M CPU @ 2.50 GHz
Memory: 7.8 GB

Gtk Theme: Ambiance
Icon Theme: ubuntu-mono-dark
Cursor Theme: DMZ-White
Window Theme: Ambiance

Sinevar



Attached are the screenshots of my favorite tweaked desktops of Ubuntu 12.04 LTS running on my laptop. I am a newbie to Ubuntu, but have heard about it before, about 6 years or so. Unfortunately, I couldn't get hands on it since I hadn't a PC of my own, but now I have one.

Dock is Cairo, Conky for monitoring, and some basic tweakings on terminal too.

Laptop : Dell Vostro 1550
Processor : Intel® Core™ i5-2430M CPU @ 2.40GHz × 4, 2nd gen.
RAM: 2 GB
HDD: 320 GB
OS : Ubuntu 12.04 LTS 64-bit

Saravanan

E	7	D	9	B	2	A	0	1	4	6	F	C	3	5	8
3	0	4	2	7	D	5	8	B	C	E	A	F	6	1	9
F	6	C	5	3	4	9	1	2	8	0	7	A	E	B	D
1	8	B	A	F	E	6	C	9	D	5	3	4	7	0	2
9	F	2	3	5	8	4	6	E	0	D	C	7	1	A	B
C	E	A	4	2	0	3	B	8	9	7	1	6	5	D	F
6	1	8	0	E	A	7	D	4	F	B	5	2	C	9	3
7	D	5	B	1	F	C	9	A	6	3	2	E	8	4	0
D	5	6	7	9	B	8	A	C	E	2	0	3	4	F	1
0	3	E	F	D	5	1	4	6	7	9	8	B	A	2	C
B	2	9	8	0	C	E	F	3	A	1	4	5	D	6	7
4	A	1	C	6	3	2	7	D	5	F	B	0	9	8	E
2	4	7	E	A	9	F	5	0	1	C	D	8	B	3	6
5	C	3	6	4	7	D	2	F	B	8	9	1	0	E	A
8	9	0	1	C	6	B	3	5	2	A	E	D	F	7	4
A	B	F	D	8	1	0	E	7	3	4	6	9	2	C	5

SUDOKU

CODE WORD

C	A	P	T	U	R	E	S		A	W	R	Y													
	L		W		E		I		D		A														
F	I	N	I	S	H		T	O	R	P	I	D													
	Q		S		E	Y	E		E		S														
J	U	S	T		A		D	O	N	Z	E	L													
	O				T				A																
S	T	O	O	L	S		P	A	L	T	R	Y													
			B				R				E														
C	R	A	V	A	T		U		F	A	C	T													
	O		I		O	L	D		L		I														
U	P	R	O	A	R		E	X	A	C	T	S													
	E		U		E		N		I		E														
A	S	K	S		S	A	T	I	R	I	S	T													
1	2	3	4	5	6	7	8	9	10	11	12	13													
H	O	V	J	Y	D	X	L	W	C	B	G	A													
14	15	16	17	18	19	20	21	22	23	24	25	26													
Q	T	I	U	E	F	Z	K	R	M	N	S	P													
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z



Codeword and 16x16 Suduko puzzles are copyright, and kindly provided by, **The Puzzle Club** - www.thepuzzleclub.com



HOW TO CONTRIBUTE

FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your opinions, desktops, stories, how-to's, reviews, and anything else you want to tell your fellow *buntu users. Send your articles to: articles@fullcirclemagazine.org

We are always looking for new articles to include in Full Circle. For help and advice please see the **Official Full Circle Style Guide**: <http://url.fullcirclemagazine.org/75d471>

FCM#71

Deadline:

Sunday 10th Mar. 2013.

Release:

Friday 29th Mar. 2013.



Send your **comments** or Linux experiences to: letters@fullcirclemagazine.org
Hardware/software **reviews** should be sent to: reviews@fullcirclemagazine.org
Questions for Q&A should go to: questions@fullcirclemagazine.org
Desktop screens should be emailed to: misc@fullcirclemagazine.org
... or you can visit our **forum** via: fullcirclemagazine.org



Full Circle Team



Editor - Ronnie Tucker

ronnie@fullcirclemagazine.org

Webmaster - Rob Kerfia

admin@fullcirclemagazine.org

Podcast - Les Pounder & Co.

podcast@fullcirclemagazine.org

Editing & Proofreading

Mike Kennedy, Lucas Westermann,
Gord Campbell, Robert Orsino,
Josh Hertel, Bert Jerred

Our thanks go to Canonical, the many translation teams around the world and **Thorsten Wilms** for the FCM logo.

Getting Full Circle Magazine:



EPUB Format - Recent editions of Full Circle have a link to the epub file on the downloads page. If you have any problems with the epub file, you can drop an email to: mobile@fullcirclemagazine.org



Google Currents - Install the Google Currents app on your Android/Apple devices, search for 'full circle' (within the app) and you'll be able to add issues 55+. Or, you can click the links on the FCM download pages.



Ubuntu Software Centre - You can get FCM via the Ubuntu Software Centre: <https://apps.ubuntu.com/cat/>. Search for 'full circle', choose an issue, and click the download button.



Issuu - You can read Full Circle online via Issuu: <http://issuu.com/fullcirclemagazine>. Please share and rate FCM as it helps to spread the word about FCM and Ubuntu Linux.



Ubuntu One - You can now have an issue delivered to your free Ubuntu One space by clicking the 'Send to Ubuntu One' button which is available on issues 51+.

