



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #52 - August 2011



REVIEW:
CHROMEBOOK



ZONEMINDER CCTV - PART 1
ZONEMINDER + WEBCAM = SECURITY CAMERA

**NEW
SERIES!**

HowTo



Program In Python Pt26 p.07

	A	B	C
1	Title	First Name	Last Name
2	Mr.	Tom	Jones
3	Dr.	Harry	McMahon
4	Rev.	Mike	Mickey

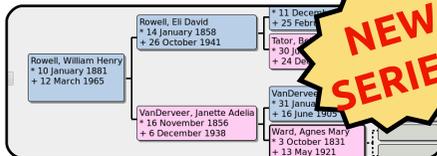
LibreOffice Pt7 p.16



Ubuntu Dev. Pt4 p.19



Business & Edu. Pt1 p.23



GRAMPS Pt.1 p.27

NEW SERIES!

NEW SERIES!



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY



Linux News p.04



My Desktop p.53

Columns

```
#An alias to make the
command more detailed
alias ls = "ls -la --
color=always --classi
```

Command & Conquer p.05



Ubuntu Games p.49



Linux Labs p.33



Q&A p.51



Ubuntu Women p.47



Write For Full Circle p.32

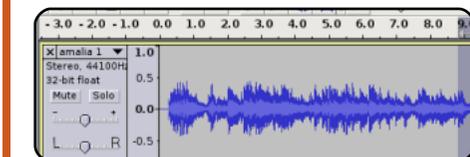
Opinions



My Story p.36



My Opinion p.37



I Think... p.39



Review p.41



Letters p.45



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!

Three new series start this month. Two of them are HowTo articles: *How To Use Ubuntu in Home, Business and Education*, and the other is on how to use the genealogy application *GRAMPS*. This means you now have five HowTo articles this month, and probably every month from now on. The third new item is in Linux Labs, and that's on my exploits with *ZoneMinder* and a webcam to create a simple CCTV system.

Although you now have five HowTo articles, I have only have enough screens for one more month of *My Desktop*, so now is a good time to submit your desktop screenshot and info. If you want, send us a photo of your actual desktop/laptop, and it's specs, and we'll do a *My PC* section too. I'm also all out of *My Opinion* articles, so feel free to email in your Ubuntu/Linux rants. Please note: no Unity rants! We've covered how most people seem to despise it, so no need to reinvent the wheel. It's that or I keep printing Robin Catling's rants. Which, now that I think about it, don't seem to provoke the same response that they used to. Is it Robin who's going soft, or you guys (and gals)?

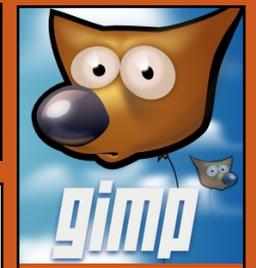
I'm also looking for feedback from you, the readers, on the magazine's contents. Is there a section you don't like, or think is pointless? Is there a section you'd like to see added? Please bear in mind though that the regular writers and myself can write only about what we know. But that's why I like it when you folks send us your articles on subjects that we don't know anything about, or rarely cover. So, don't be shy, send us your articles about anything Ubuntu or Linux related. The more unusual and obscure the better! As long as you give it a quick second read, and double check your spelling, the awesome FCM proof-readers will edit it for you.

All the best, and keep in touch.

Ronnie

ronnie@fullcirclemagazine.org

This magazine was created using :



Full Circle Podcast

Released every two weeks, 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:

Robin Catling

Ed Hewitt

Dave Wilkins

<http://fullcirclemagazine.org>





Linux Mint 11 Debian Edition

The [Mint] team is proud to announce the release of LMDE 201108 RC with updated ISOs for Gnome and Xfce.

Highlights

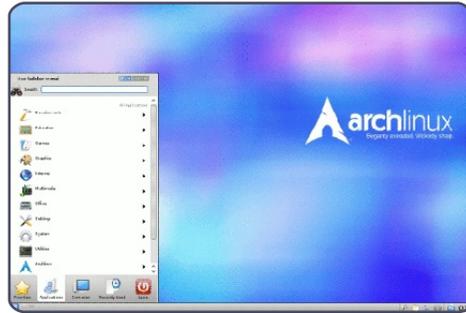
- All Linux Mint 11 features
- Installer improvements (keyboard variants, locale, bug fixes, UUID in fstab)
- Update Packs, dedicated Update Manager and staged repositories
- GTK2/GTK3 theme compatibility
- Updated software and packages

Linux Mint Debian Edition (LMDE) is a rolling distribution based on Debian Testing.

It's available in both 32 and 64-bit as a live DVD with Gnome or Xfce. The purpose of LMDE is to look identical to the main edition and to provide the same functionality while using Debian as a base.

Source: blog.linuxmint.com

Arch Linux Moves Up To Linux 3.0



The Arch Linux team has released the first new all-in-one update for its minimalist, rolling-release distribution in 15 months. The Arch Linux 2011.08.19 installation media features support for Linux 3.0 and the syslinux bootloader, and offers experimental Btrfs and NILFS2 file-systems, and more flexible source-file selection.

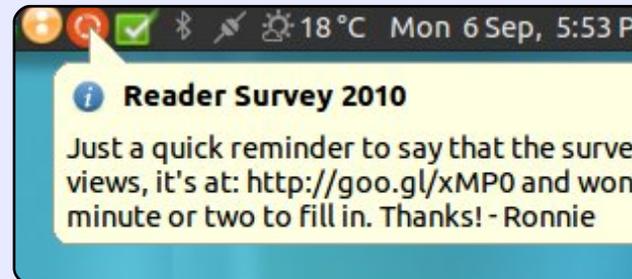
Arch Linux is a rolling-release distro (as is Gentoo), making packages available to the distribution shortly after they are released upstream. Since Arch Linux doesn't draw attention to itself with release announcements, it's easy to forget about it.

Source: desktoplinux.com

Full Circle Notifier

Our very own **Full Circle Notifier** is now at 1.0.2. FCN is a small application that sits in your system tray and will not only announce issue/podcast releases, and can be set to automatically download them for you too! Several people have created packages of FCN and translations are starting.

For more info, see the FCN Google Group:
<http://goo.gl/4Ob4>



Ubuntu One Hits 1 Million Users



Fresh off the back of our super successful

Android Files app launch, we now get to share even more exciting news. Back in May we surpassed the 1 million global user mark! This is a significant milestone for us at Ubuntu One.

We're passionate about building great Ubuntu One services and seeing how people love using them, so we'd like to say a big thank you to all our loyal users for supporting Ubuntu One. As such, we have made some changes to our plans. From today, Ubuntu One Basic will become Ubuntu One Free and users will get 5GB of free storage when they set up an account. Current users of Ubuntu One Basic will see their free storage allowance automatically increase from 2GB to 5GB.

Source: ubuntulook.com



I recently received an email from a reader, Chris, who can be found on launchpad.net as MrChris. He was asking me about Japanese input in LaTeX, as a follow-up to my article in FCM Issue #50. I didn't know the answer off the top of my head, but, together, we figured out a solution, which I felt I should share with anyone who may have the same question. Also, since the latex-cjk-* packages stand for "Chinese, Japanese, Korean", I've covered (roughly) how to get each of the 3 languages working.

Packages:
`texlive`
`texlive-latex-extra`
`latex-cjk-common`
`latex-cjk-japanese`
`latex-cjk-japanese-wadalab`

For Chinese (using Method #1 listed below), you'll also need:
`latex-cjk-chinese`
`latex-cjk-chinese-arphic-gkai00mp`

For Japanese, I used the following template:
<http://pastebin.com/tasDkhZ3>

As a brief explanation to the code:

`\usepackage{ucs}` – unicode support
`\usepackage[utf8x]{inputenc}` – extended UTF-8 encoding (includes asian characters)
`\usepackage[english]{babel}` – sets the typographical rules (in this case, to English). This includes, among others, where to break words if it flows over a line. By assigning a typographical set of rules that doesn't apply to Asian characters, you ensure that groups of Kanji aren't separated when breaking over a line. In using a language that doesn't support asian locale, you ensure that Kanji that need to be grouped together, aren't split.

`\usepackage[overlap, CJK]{ruby}` – This is the package required to get CJK working
`\usepackage{CJKulem}` – Used for certain options for CJK

`\renewcommand{\rubysep}{0.2ex}` – changes the default

spacing
`\newenvironment{Japanese}{\CJKfamily{min}\CJKtilde\CJKnospace}}` - This creates a new environment (like "document", "enumerate", "itemize", etc.), which configures the options for Japanese input.

The rest of the code is pretty self-explanatory. Once you compile the file, the PDF should contain the Japanese, in a nicely-formatted font. The only issue is that the text is not compiled top-to-bottom and right-to-left, as is the standard for Japanese. I haven't yet found a solution that works well for this. If anyone has a suggestion, feel free to pass it along.

For Chinese:

Method #1:

The same template that works for Japanese would work for Chinese too, with the following adjustments:

`\newenvironment{Chinese}{`

`\CJKfamily{gkai}`
`\CJKtilde`
`\CJKnospace}}`

Also, the line "`\begin{Japanese}`" would logically have to be changed to "`\begin{Chinese}`". Besides that, it will compile. As mentioned above, this will result in left-to-right and horizontal text. For top-to-bottom and right-to-left formatting, move on to Method #2.

Method #2:

Disclaimer: I have not tried this method, nor do I know who originally wrote the script. However, Chris, the reader who originally posed the question, has tried it and gotten it to work, so I felt it deserved to be included here.

Download the following script:
http://scripts.sil.org/cms/scripts/request_download.php?&format=file&media_id=xetex_chinese_sample&filename=xetex_chinese_sample.zip

According to Chris, he had to

make the following changes:

“Unfortunately, the uploader created it on a Mac where the fonts are different.

However, in XeTeX you can specify fonts that are available in other programs and system-wide - so you can just pick a nice font in LibreOffice and specify that one. After changing the fonts, using "xetex <file>.tex" [it] actually compiles successfully! See <http://pastebin.com/WiiXsbNz> for a modified file.”

For Korean, the same method as used for Japanese can be used. You'd simply need to pick a font family for Korean, and install the following package:

latex-cjk-korean

Since I don't speak Korean, I haven't been able to try it out and find the correct font family, but I would suggest trying it without a font family specified, or else searching for a test document online.

I hope you have found this article useful. If anyone has a specific issue (or question), you can reach me at

lswest34@gmail.com. Please be sure to include “FCM” or “C&C” in the subject line, so that I don't overlook it. Any suggestions, or corrections, can also be emailed to me at the above address. If your enquiry requires a large excerpt of code, I would appreciate it if you would upload it to pastebin.com, or some other such site, and simply supply the link in the email. If the code isn't too long, please quote it or clearly denote the text from the email, in order to allow for quick referencing.

References:

Original Japanese test document is from here:

<https://bbs.archlinux.org/viewtopic.php?pid=622937#p622937>

Information on UCS is from here:

<http://kile.sourceforge.net/Documentation/html/lang.html#ucs>

A blog post by Chris on Japanese:

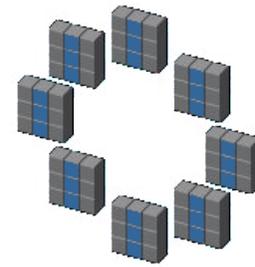
<http://blog.mindfall.net/?p=221>

A blog post by Chris on Chinese:

<http://blog.mindfall.net/?p=236>



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.



ServerCircle

Server Circle is a new question and answer site run by techies.

Users with any level of experience can ask technical questions for free about anything server related, and receive answers from trusted experts, who are rated by the community.

With time you can earn reputation points, and even financial rewards, by contributing your answers to questions from other people.

<http://www.servercircle.com>



Recent	Popular	Unanswered	Rewards	
Which packages can I safely uninstall on Ubuntu Server ?				17 views 0 replies
answer now				
by Squeeze (445 points) in Ubuntu Linux - 0 votes				
How to access a Intel Express 535T Switch Hub				34 views 4 replies
answer now				
by GoldAlchemist (50 points) in Networking - 1 votes				
How to use wired desktop for server to wireless notebook				38 views 2 replies
answer now				
by t0xihman (50 points) in Linux Servers - 1 votes				
Help with mod_security in Apache please				

NOTE: Server Circle is not affiliated with, nor endorsed by, Full Circle magazine.



Last month we discussed tkinter and four of the widgets available: TopLevel, Frames, Buttons, and Labels. I also told you last month, I'd discuss how to have a widget as a parent other than the Toplevel widget.

So, this month, we'll discuss more on Frames, Buttons, and Labels, and introduce Checkboxes, Radio buttons, Textboxes (Entry widgets), Listboxes with a vertical scrollbar, and Messageboxes. Before we get started, let's examine some of these widgets.

Checkboxes are considered a many of many type selection widget that has two options, checked or not checked, or you could consider it on or off. They are usually used to provide a series of options where any, many, or all of those options may be selected. You can set an event to inform you when the checkbox has been toggled, or just query the value of the widget at any time.

Radiobuttons are considered a one of many type selection

widget. It also has two options, on and off. However, they are grouped together to provide a set of options that logically can have only one selection. You can have multiple groups of Radiobuttons that, if properly programmed, won't interact with each other.

A Listbox provides a list of items for the user to select from. Most times, you want the user to select only one of the items at a time, but there can be occasions that you will allow the user to select multiple items. A scroll bar can be placed either horizontally or vertically to allow the user to easily look through all the items available.

Our project will consist of a main window and seven main frames that visually group our widget sets:

- The first frame will be very basic. It simply consists of various labels, showing the different relief options.
- The second will contain buttons, again pretty simple, that

use the different relief options.

- In this frame, we'll have two checkboxes and a button that can programmatically toggle them, and they will send their state (1 or 0) back to the terminal window when clicked or toggled.

- Next, we'll have two groups of three radio buttons, each sending a message to the terminal window when clicked. Each group is separate.

- This has some text or entry boxes, which aren't new to you, but there's also a button to enable and disable one of them. When disabled, no entry can be made to that textbox.

- This is a list box with a vertical scroll bar that sends a message to the terminal whenever an item is selected, and will have two buttons. One button will clear the list box and the other will fill it with some dummy values.

- The final frame will have a series of buttons that will call

```
# widgetdemol.py
# Labels
from Tkinter import *

class Demo:
    def __init__(self, master):
        self.DefineVars()
        f = self.BuildWidgets(master)
        self.PlaceWidgets(f)
```

various types of message boxes.

So now, we'll start our project. Let's name it "widgetdemo1.py". Be sure to save it because we will be writing our project in little pieces, and build on them to make our full app. Each piece revolves around one of the frames. You'll notice that I'm including a number of comments as we go, so you can refer back to what's happening. Above are first few lines.

The first two lines (comments) are the name of the application and what we are concentrating on in this part. Line three is our import statement. Then we define our class. The next line starts our `__init__` routine, which you all should be familiar with by now,

but, if you are just joining us, it's the code that gets run when we instantiate the routine in the main portion of the program. We are passing it the Toplevel or root window, which comes in as master here. The last three lines (so far), call three different routines. The first (DefineVars) will set up various variables we'll need as we go. The next (BuildWidgets) will be where we define our widgets, and the last (PlaceWidgets) is where we actually place the widgets into the root window. As we did last time, we'll be using the grid geometry manager. Notice that BuildWidgets will return the object "f" (which is our root window), and we'll pass that along to the PlaceWidgets routine.

Above right is our BuildWidgets routine. Each of the lines that start with "self." have been split for two reasons. First, it's good practice to keep the line length to 80 characters or less. Secondly, it makes it easier on our wonderful editor. You can do two things. One, just make each line long, or keep it as is. Python lets us split lines as long as they are within parentheses or brackets. As I said earlier, we are defining the widgets before we place them in

```
def BuildWidgets(self, master):
    # Define our widgets
    frame = Frame(master)
    # Labels
    self.lblframe = Frame(frame, relief = SUNKEN, padx = 3, pady = 3,
                          borderwidth = 2, width = 500)
    self.lbl1 = Label(self.lblframe, text="Flat Label", relief = FLAT,
                     width = 13, borderwidth = 2)
    self.lbl2 = Label(self.lblframe, text="Sunken Label", relief = SUNKEN,
                     width = 13, borderwidth = 2)
    self.lbl3 = Label(self.lblframe, text="Ridge Label", relief = RIDGE, width = 13,
                     borderwidth = 2)
    self.lbl4 = Label(self.lblframe, text="Raised Label", relief = RAISED,
                     width = 13, borderwidth = 2)
    self.lbl5 = Label(self.lblframe, text="Groove Label", relief = GROOVE,
                     width = 13, borderwidth = 2)

    return frame
```

the grid. You'll notice when we do the next routine, that we can also define a widget at the time we place it in the grid, but defining it before we put it in the grid in a routine like this makes it easier to keep track of everything, since we are doing (most of) the definitions in this routine.

So, first we define our master frame. This is where we will be putting the rest of our widgets. Next, we define a child (of the master frame) frame that will hold five labels, and call it lblframe. We set the various attributes of the frame here. We set the relief to 'SUNKEN', a padding of 3 pixels on

left and right (padx), and 3 pixels on the top and bottom (pady). We also set the borderwidth to 2 pixels so that its sunken relief is noticeable. By default, the borderwidth is set to 0, and the effect of being sunken won't be noticed. Finally, we set the total width of the frame to 500 pixels.

Next, we define each label widget that we will use. We set the parent as self.lblframe, and not to frame. This way all the labels are children of lblframe, and lblframe is a child of frame. Notice that each definition is pretty much the same for all five of the labels except the name of the widget

(lbl1, lbl2, etc), the text, and the relief or visual effect. Finally, we return the frame back to the calling routine (`__init__`).

The following page (top right) shows our PlaceWidgets routine.

We get the frame object in as a parameter called master. We assign that to 'frame' to simply be consistent with what we did in the BuildWidgets routine. Next, we set our main grid up (`frame.grid(column = 0, row = 0)`). If we don't do this, nothing works correctly. Then we start putting our widgets into the grid locations. First we put the frame (lblframe)

that holds all our labels, and set its attributes. We put it in column 0, row 1, set the padding to 5 pixels on all sides, tell it to span 5 columns (left and right), and finally use the “sticky” attribute to force the frame to expand fully to the left and right (“WE”, or West and East). Now comes the part that sort of breaks the rule that I told you about. We are placing a label as the first widget in the frame, but we didn't define it ahead of time. We define it now. We set the parent to lblframe, just like the other labels. We set the text to “Labels |”, the width to 15, and the anchor to east ('e'). If you remember from last time, using the anchor attribute, we can set where in the widget the text will display. In this case, it's along the right border. Now the fun part. Here we define the grid location (and any other grid attributes we need to), simply by appending “.grid” at the end of the label definition.

Next, we lay out all of our other labels in the grid - starting at column 1, row 0.

Here is our DefineVars routine. Notice that we simply use the pass

statement for now. We'll be filling it in later on, and we don't need it for this part:

```
def DefineVars(self):  
    # Define our  
    resources  
    pass
```

And lastly we put in our main routine code:

```
root = Tk()  
root.geometry('750x40+150+150')  
root.title("Widget  
Demo 1")  
demo = Demo(root)  
root.mainloop()
```

First, we instantiate an instance of Tk. Then we set the size of the main window to 750 pixels wide by 40 pixels high, and locate it at 150 pixels from the left and top of the screen. Then we set the title of the window and instantiate our Demo object, and finally call the Tk mainloop.

Give it a try. You should see the five labels plus the “last minute” label in various glorious effects.

Buttons

Now save what you have as

```
def PlaceWidgets(self, master):  
    frame = master  
    # Place the widgets  
    frame.grid(column = 0, row = 0)  
    # Place the labels  
    self.lblframe.grid(column = 0, row = 1, padx = 5, pady = 5,  
                        columnspan = 5, sticky='WE')  
    l = Label(self.lblframe, text='Labels |', width=15,  
              anchor='e').grid(column=0, row=0)  
    self.lbl1.grid(column = 1, row = 0, padx = 3, pady = 5)  
    self.lbl2.grid(column = 2, row = 0, padx = 3, pady = 5)  
    self.lbl3.grid(column = 3, row = 0, padx = 3, pady = 5)  
    self.lbl4.grid(column = 4, row = 0, padx = 3, pady = 5)  
    self.lbl5.grid(column = 5, row = 0, padx = 3, pady = 5)
```

widgetdemo1a.py, and let's add some buttons. Since we built our base program to be added to, we'll simply add the parts that apply. Let's start with the BuildWidgets routine. After the labels definitions, and before the “return frame” line, add what is shown on the next page, top right.

Nothing really new here. We've defined the buttons, with their

attributes, and set their callbacks via the .bind configuration. Notice that we are using lambda to send the values 1 through 5 based on which button is clicked. In the callback, we'll use that so we know which button we are dealing with. Now we'll work in the PlaceWidgets routine. Put the code below after the last label placement.

```
# Place the buttons  
self.btnframe.grid(column=0, row = 2, padx = 5,  
                    pady = 5, columnspan = 5, sticky = 'WE')  
l = Label(self.btnframe, text='Buttons |', width=15,  
          anchor='e').grid(column=0, row=0)  
self.btn1.grid(column = 1, row = 0, padx = 3, pady = 3)  
self.btn2.grid(column = 2, row = 0, padx = 3, pady = 3)  
self.btn3.grid(column = 3, row = 0, padx = 3, pady = 3)  
self.btn4.grid(column = 4, row = 0, padx = 3, pady = 3)  
self.btn5.grid(column = 5, row = 0, padx = 3, pady = 3)
```

Once again, nothing really new here, so we'll move on. Bottom right is our callback routine. Put it after the DefineVars routine.

Again, nothing really fancy here. We just use a series of IF/ELIF routines to print what button was clicked. The main thing to look at here (when we run the program) is that the sunken button doesn't "move" when you click on it. You would not usually use the sunken relief unless you were making a button that stays "down" when you click it. Finally, we need to tweak the geometry statement to support the extra widgets we put in:

```
root.geometry('750x110+150+150')
```

Ok. All done with this one. Save it and run it.

Now save this as widgetdemo1b.py, and we'll move on to checkboxes.

Checkboxes

As I said earlier, this part of the demo has a normal button and two

checkboxes. The first checkbox is what you would normally expect a checkbox to look like. The second is more like a "sticky" button - when it's not selected (or checked), it looks like a normal button. When you select it, it looks like a button that is stuck down. We can do this by simply setting the indicatoron attribute to False. The "normal" button will toggle the checkboxes from checked to unchecked, and vice versa, each time you click the button. We get to do this programmatically by calling the .toggle method attached to the checkbox. We bind the left mouse button click event (button release) to a function so we can send a message (in this case) to the terminal. In addition to all of this, we are setting two variables (one for each of the checkboxes) that we can query at any time. In this case, each time the checkbox is clicked we query this value and print it. Pay attention to the variable portion of the code. It is

```
# Buttons
self.btnframe = Frame(frame,relief = SUNKEN,padx = 3, pady = 3,
                       borderwidth = 2, width = 500)
self.btn1 = Button(self.btnframe,text="Flat Button",
                   relief = FLAT, borderwidth = 2)
self.btn2 = Button(self.btnframe,text="Sunken Button",
                   relief = SUNKEN, borderwidth = 2)
self.btn3 = Button(self.btnframe,text="Ridge Button",
                   relief = RIDGE, borderwidth = 2)
self.btn4 = Button(self.btnframe,text="Raised Button",
                   relief = RAISED, borderwidth = 2)
self.btn5 = Button(self.btnframe,text="Groove Button",
                   relief = GROOVE, borderwidth = 2)
self.btn1.bind('<ButtonRelease-1>',lambda e: self.BtnCallback(1))
self.btn2.bind('<ButtonRelease-1>',lambda e: self.BtnCallback(2))
self.btn3.bind('<ButtonRelease-1>',lambda e: self.BtnCallback(3))
self.btn4.bind('<ButtonRelease-1>',lambda e: self.BtnCallback(4))
self.btn5.bind('<ButtonRelease-1>',lambda e: self.BtnCallback(5))
```

```
def BtnCallback(self,val):
    if val == 1:
        print("Flat Button Clicked...")
    elif val == 2:
        print("Sunken Button Clicked...")
    elif val == 3:
        print("Ridge Button Clicked...")
    elif val == 4:
        print("Raised Button Clicked...")
    elif val == 5:
        print("Groove Button Clicked...")
```

used in many widgets.

Under the BuildWidget routine, after the button code we just put in and before the return statement, put the code shown on the next page, top right.

Again, you have seen all of this before. We create the frame to hold our widgets. We set up a button and two check boxes. Let's place them now using the code on the next page, middle right.

Now we define the two

HOWTO - PROGRAM IN PYTHON - PART 26

variables that we will use to monitor the value of each check box. Under DefineVars, comment out the pass statement, and add this...

```
self.Chk1Val = IntVar()  
self.Chk2Val = IntVar()
```

After the button callback return, put the text shown bottom right.

And finally replace the geometry statement with this:

```
root.geometry('750x170+150+150')
```

Save and run. Save it as widgetdemo1c.py, and let's do radio buttons.

Radiobuttons

If you are old enough to remember car radios with push buttons to select the station presets, you'll understand why these are called Radiobuttons. When using radiobuttons, the variable attribute is very important. This is what groups the radiobuttons together. In this demo, the first group of buttons is grouped by the variable named

self.RBVal. The second is grouped by the variable self.RBValue2. We also need to set the value attribute at design time. This ensures that the buttons will return a value that makes sense whenever they are clicked.

Back to BuildWidgets, and, just before the return statement, add the code shown on the following page.

One thing of note here. Notice the "last minute" label definitions in the

```
# Check Boxes  
self.cbframe = Frame(frame, relief = SUNKEN, padx = 3, pady = 3,  
                      borderwidth = 2, width = 500)  
self.chk1 = Checkbutton(self.cbframe, text = "Normal Checkbox",  
                        variable=self.Chk1Val)  
self.chk2 = Checkbutton(self.cbframe, text = "Checkbox",  
                        variable=self.Chk2Val, indicatoron = False)  
self.chk1.bind('<ButtonRelease-1>', lambda e: self.ChkBoxClick(1))  
self.chk2.bind('<ButtonRelease-1>', lambda e: self.ChkBoxClick(2))  
self.btnToggleCB = Button(self.cbframe, text="Toggle Cbs")  
self.btnToggleCB.bind('<ButtonRelease-1>', self.btnToggle)
```

```
# Place the Checkboxes and toggle button  
self.cbframe.grid(column = 0, row = 3, padx = 5, pady = 5,  
                  columnspan = 5, sticky = 'WE')  
l = Label(self.cbframe, text='Check Boxes |', width=15,  
          anchor='e').grid(column=0, row=0)  
self.btnToggleCB.grid(column = 1, row = 0, padx = 3, pady = 3)  
self.chk1.grid(column = 2, row = 0, padx = 3, pady = 3)  
self.chk2.grid(column = 3, row = 0, padx = 3, pady = 3)
```

```
def btnToggle(self, p1):  
    self.chk1.toggle()  
    self.chk2.toggle()  
    print("Check box 1 value is {}".format(self.Chk1Val.get()))  
    print("Check box 2 value is {}".format(self.Chk2Val.get()))  
  
def ChkBoxClick(self, val):  
    if val == 1:  
        print("Check box 1 value is {}".format(self.Chk1Val.get()))  
    elif val == 2:  
        print("Check box 2 value is {}".format(self.Chk2Val.get()))
```

HOWTO - PROGRAM IN PYTHON - PART 26

PlaceWidget routine. These long lines are broken up to show how to use parens to allow our long lines to be formatted nicely in our code, and still function correctly.

In DefineVars add:

```
self.RBVal = IntVar()
```

Add the click routines:

```
def RBClick(self):
```

```
    print("Radio Button  
clicked - Value is  
{0}".format(self.RBVal.get()))
```

```
def RBClick2(self):
```

```
    print("Radio Button  
clicked - Value is  
{0}".format(self.RBVal2.get()))
```

```
# Radio Buttons
self.rbframe = Frame(frame, relief = SUNKEN, padx = 3, pady = 3, borderwidth = 2, width = 500)
self.rb1 = Radiobutton(self.rbframe, text = "Radio 1", variable = self.RBVal, value = 1)
self.rb2 = Radiobutton(self.rbframe, text = "Radio 2", variable = self.RBVal, value = 2)
self.rb3 = Radiobutton(self.rbframe, text = "Radio 3", variable = self.RBVal, value = 3)
self.rb1.bind('<ButtonRelease-1>', lambda e: self.RBClick())
self.rb2.bind('<ButtonRelease-1>', lambda e: self.RBClick())
self.rb3.bind('<ButtonRelease-1>', lambda e: self.RBClick())
self.rb4 = Radiobutton(self.rbframe, text = "Radio 4", variable = self.RBVal2, value = "1-1")
self.rb5 = Radiobutton(self.rbframe, text = "Radio 5", variable = self.RBVal2, value = "1-2")
self.rb6 = Radiobutton(self.rbframe, text = "Radio 6", variable = self.RBVal2, value = "1-3")
self.rb4.bind('<ButtonRelease-1>', lambda e: self.RBClick2())
self.rb5.bind('<ButtonRelease-1>', lambda e: self.RBClick2())
self.rb6.bind('<ButtonRelease-1>', lambda e: self.RBClick2())
```

In PlaceWidgets, add this:

```
# Place the Radio Buttons and select the first one
self.rbframe.grid(column = 0, row = 4, padx = 5, pady = 5, columnspan = 5, sticky = 'WE')
l = Label(self.rbframe,
          text='Radio Buttons |',
          width=15, anchor='e').grid(column=0, row=0)
self.rb1.grid(column = 2, row = 0, padx = 3, pady = 3, sticky = 'EW')
self.rb2.grid(column = 3, row = 0, padx = 3, pady = 3, sticky = 'WE')
self.rb3.grid(column = 4, row = 0, padx = 3, pady = 3, sticky = 'WE')
self.RBVal.set("1")
l = Label(self.rbframe, text='| Another Set |',
          width = 15,
          anchor = 'e').grid(column = 5, row = 0)
self.rb4.grid(column = 6, row = 0)
self.rb5.grid(column = 7, row = 0)
self.rb6.grid(column = 8, row = 0)
self.RBVal2.set("1-1")
```

and, finally, rework the geometry statement as follows.

```
root.geometry('750x220+150+150')
```

Save the project as widgetdemo1d.py, and run it. Now, we'll start working on standard textboxes (or entry widgets).

Entry

Again, we've used textboxes or entry widgets in various GUI flavors before. However this time, as I said earlier, we will show how to keep the user from making changes to the textbox by disabling it. This is helpful if you are showing some data, and allowing the user to change it only when in the "edit" mode. By now, you should be pretty sure that the first thing we need to do is add code (shown right) to the BuildWidget routine.

Listbox

Next we'll work our listbox. Starting in BuildWidgets, add the code from the following page, right side.

As usual, we create our frame. Then we create our vertical scroll bar. We do this before we create the list box, because we have to reference the scrollbar '.set' method. Notice the attribute 'height = 5'. This forces the listbox to show 5 items at a time. In the .bind statement, we use '<<ListboxSelect>>' as the event. It's called a virtual event, since it's not really an "official" event.

Now, we'll deal with the additional code for the PlaceWidgets routine, and that's shown on the following page, left side.

Message Dialogs

This section is

```
# Textboxes
self.tbframe = Frame(frame, relief = SUNKEN, padx = 3, pady = 3, borderwidth = 2, width = 500)
self.txt1 = Entry(self.tbframe, width = 10)
self.txt2 = Entry(self.tbframe, disabledbackground="#cccccc", width = 10)
self.btnDisable = Button(self.tbframe, text = "Enable/Disable")
self.btnDisable.bind('<ButtonRelease-1>', self.btnDisableClick)
```

Next, add this code to the PlaceWidget routine:

```
# Place the Textboxes
self.tbframe.grid(column = 0, row = 5, padx = 5, pady = 5, columnspan = 5, sticky = 'WE')
l = Label(self.tbframe, text='Textboxes |', width=15, anchor='e').grid(column=0, row=0)
self.txt1.grid(column = 2, row = 0, padx = 3, pady = 3)
self.txt2.grid(column = 3, row = 0, padx = 3, pady = 3)
self.btnDisable.grid(column = 1, row = 0, padx = 3, pady = 3)
```

Add this line to the bottom of the DefineVars routine:

```
self.Disabled = False
```

Now, add the function that responds to the button click event:

```
def btnDisableClick(self, p1):
    if self.Disabled == False:
        self.Disabled = True
        self.txt2.configure(state='disabled')
    else:
        self.Disabled = False
        self.txt2.configure(state='normal')
```

And finally, rework the geometry statement:

```
root.geometry('750x270+150+150')
```

Save it as widgetdemo1d.py, and run it.

```
# Place the Listbox and support buttons
self.lstframe.grid(column = 0, row = 6, padx = 5,
pady = 5, columnspan = 5, sticky = 'WE')
l = Label(self.lstframe, text='List Box |', width=15,
anchor='e').grid(column=0, row=0, rowspan=2)
self.lbox.grid(column = 2, row = 0, rowspan=2)
self.VScroll.grid(column = 3, row = 0, rowspan = 2,
sticky = 'NSW')
self.btnClearLBox.grid(column = 1, row = 0, padx =
5)
self.btnFillLBox.grid(column = 1, row = 1, padx = 5)
```

In DefineVars add this...

```
# List for List box items
self.examples = ['Item One', 'Item Two', 'Item
Three', 'Item Four']
```

And add the following support routines:

```
def ClearList(self):
    self.lbox.delete(0,END)

def FillList(self):
    # Note, clear the listbox first...no check is done
    for ex in self.examples:
        self.lbox.insert(END,ex)
    # insert([0,ACTIVE,END],item)

def LBoxSelect(self,p1):
    print("Listbox Item clicked")
    items = self.lbox.curselection()
    selitem = items[0]
    print("Index of selected item =
{0}".format(selitem))
    print("Text of selected item =
{0}".format(self.lbox.get(selitem)))
```

Finally, update the geometry line.

```
root.geometry('750x370+150+150')
```

Save this as widgetdemo1e.py, and run it. Now we will do our last modifications to our application.

```
# List Box Stuff
self.lstframe = Frame(frame,
relief = SUNKEN,
padx = 3,
pady = 3,
borderwidth = 2,
width = 500
)
# Scrollbar for list box
self.VScroll = Scrollbar(self.lstframe)
self.lbox = Listbox(self.lstframe,
height = 5,
yscrollcommand = self.VScroll.set)
# default height is 10
self.lbox.bind('<<ListboxSelect>>',self.LBox
Select)
self.VScroll.config(command =
self.lbox.yview)
self.btnClearLBox = Button(
self.lstframe,
text = "Clear List",
command = self.ClearList,
width = 11
)
self.btnFillLBox = Button(
self.lstframe,
text = "Fill List",
command = self.FillList,
width = 11
)
# <<ListboxSelect>> is virtual event
# Fill the list box
self.FillList()
```

simply a series of “normal” buttons that will call various types of Message Dialogs. We've done them before in a different GUI toolkit. We will explore only 5 different types, but there are more. In this section, we'll look at Info, Warning, Error, Question, and Yes/No dialogs. These are very useful when you need to pass some information to your user in a rather big way. In the BuildWidgets routine add the code shown below.

Here is the support routine. For the first three (Info, Warning, and Error), you simply call 'tkMessageBox.showinfo', or whichever you need, with two parameters. First is the title for the message dialog, and second is the actual message you want to show. The icon is handled for you by tkinter. For the dialogs that provide a response (question, yes/no), we provide a variable that receives the value of which button was clicked. In the case of the question dialog, the response is either “yes” or “no”, and, in the case of the yes/no dialog, the response is either “True” or “False”.

Finally, modify the geometry line:

```
root.geometry('750x490+550+150')
```

Save this as widgetdemo1f.py, and play away.

I've put the code for widgetdemo1f.py on pastebin at <http://pastebin.com/ZqrqHcdG>.

```
def ShowMessageBox(self,which):
    if which == 1:
        tkMessageBox.showinfo('Demo','This is an INFO messagebox')
    elif which == 2:
        tkMessageBox.showwarning('Demo','This is a WARNING messagebox')
    elif which == 3:
        tkMessageBox.showerror('Demo','This is an ERROR messagebox')
    elif which == 4:
        resp = tkMessageBox.askquestion('Demo','This is a QUESTION messagebox?')
        print('{0} was pressed...'.format(resp))
    elif which == 5:
        resp = tkMessageBox.askyesno('Demo','This is a YES/NO messagebox')
        print('{0} was pressed...'.format(resp))
```

```
# Buttons to show message boxes and dialogs
self.mbframe = Frame(frame,relief = SUNKEN,padx = 3, pady = 3, borderwidth = 2)
self.btnMBInfo = Button(self.mbframe,text = "Info")
self.btnMBWarning = Button(self.mbframe,text = "Warning")
self.btnMBError = Button(self.mbframe,text = "Error")
self.btnMBQuestion = Button(self.mbframe,text = "Question")
self.btnMBYesNo = Button(self.mbframe,text = "Yes/No")
self.btnMBInfo.bind('<ButtonRelease-1>', lambda e: self.ShowMessageBox(1))
self.btnMBWarning.bind('<ButtonRelease-1>', lambda e: self.ShowMessageBox(2))
self.btnMBError.bind('<ButtonRelease-1>', lambda e: self.ShowMessageBox(3))
self.btnMBQuestion.bind('<ButtonRelease-1>', lambda e: self.ShowMessageBox(4))
self.btnMBYesNo.bind('<ButtonRelease-1>', lambda e: self.ShowMessageBox(5))
```

Now, add the code for the PlaceWidgets routine:

```
# Messagebox buttons and frame
self.mbframe.grid(column = 0,row = 7, columnspan = 5, padx = 5, sticky = 'WE')
l = Label(self.mbframe,text='Message Boxes |',width=15, anchor='e').grid(column=0,row=0)
self.btnMBInfo.grid(column = 1, row = 0, padx= 3)
self.btnMBWarning.grid(column = 2, row = 0, padx= 3)
self.btnMBError.grid(column = 3, row = 0, padx= 3)
self.btnMBQuestion.grid(column = 4, row = 0, padx= 3)
self.btnMBYesNo.grid(column = 5, row = 0, padx= 3)
```



Until now we have been working in LibreOffice Writer, but today we will step away for just a second to do something a little different. We're going to create a poor man's database in Calc and then return to Writer and create a mail merge. This will give you some ideas about how you can use a spreadsheet as a database. In order to do a mail merge, we need to work with some sort of database. We could create our own in Base, but for a simple address book, Calc is an easier solution.

We will start by opening LibreOffice and clicking on Spreadsheet. This will open Calc with a blank spreadsheet. We won't go into a lot of details about the layout of Calc; we'll save that

for another day. What you do need to know is that a spreadsheet is laid out as a series of columns and rows. The columns are labeled above with letters (A,B,C,...) and the rows are down the left side labeled with numbers (1,2,3,...). When we use Calc as a database, the first row must contain the field titles for the database. These are the database field headers. We will put the following field headers in the first row: Title, First Name, Last Name, Street Address, City, State, Postal Code, and Country. Each field must be in a separate column starting at column A, so put the cursor in column A of the first row and type in Title. You can use the right arrow key on the keyboard to move to the next column, and fill in the rest of the fields.

	A	B	C	D	E	
1	Title	First Name	Last Name	Street Address	City	S
2	Mr.	Tom	Jones	123 Some St.	Someplace	NC
3	Dr.	Harry	McMahon	321 No Road	Noplace	NY
4	Rev.	Mike	Mickey	547 Trinity Way	Gospel	CT
5	Ms.	Amber	Sams	54 This Way	Thatplace	CA
6	Mr.	Charlie	Hacker	101 Binary Way	Lunix	NJ
7	Ms.	Molly	Holly	22 Harvard Rd	Wally	SD
8						
9						

What do you want to do?

- Create a new database
- Open an existing database file

Recently used

Addresses

Open...

- Connect to an existing database

Spreadsheet

Set up a connection to spreadsheets

Click 'Browse...' to select a LibreOffice spreadsheet or Microsoft Excel workbook. LibreOffice will open this file in read-only mode.

Location and file name

/home/elmer/Documents/Writing/LibreOffice Series/Poorm

Browse

Password required

Decide how to proceed after saving the database

Do you want the wizard to register the database in LibreOffice?

- Yes, register the database for me
- No, do not register the database

After the database file has been saved, what do you want to do?

- Open the database for editing
- Create tables using the table wizard

Starting with the second row, we begin to fill in our data. Each row is a record. Think of a record as all the information on one person. For our purposes today, you don't need to worry too much if the data is wider than the columns. If you want to adjust the width of the columns while you input your information, just place your mouse over the line separating the two columns. The cursor will become a double arrow, and you can click and drag to increase or decrease the width of the column. Once we have entered all our data, we need to save the spreadsheet. I saved mine as Poorman_db.ods. Once it is saved, close it.

Before we can use our poor man's database, we need to register it in LibreOffice. File > New > Database will bring up the database wizard. Select "connect to an existing database", and in the drop-down box, select spreadsheet. Click the Next button. Use the Browse button to find your spreadsheet. Click Next. Make sure "Yes, register the database for me" is selected, and uncheck "Open database for editing." Click Finish, and give the database a name (I named mine

Addresses). The database you just created is linked to the spreadsheet. Any updates you make to the spreadsheet will show in the database. The only catch is you cannot edit the spreadsheet when you have Writer opened. Basically, your spreadsheet database can be the only LibreOffice document open. Otherwise, LibreOffice will open the spreadsheet in read-only mode.

Now that we have created our database, and registered it in LibreOffice, we can use it to create a mail-merge. Mail-merge is great for sending form letters to customers or prospects. The cool thing is you can put any kind of data you want in your database and then use it in the letter. For example, if you have a rewards program for your customers, you could have a field for points, and include the points each customer has earned in the letter, but you would have to type the letter only once.



Click on the Data Sources icon (shown left), View > Data sources, or press F4. This will display the registered databases below the

Drag field headers into the document

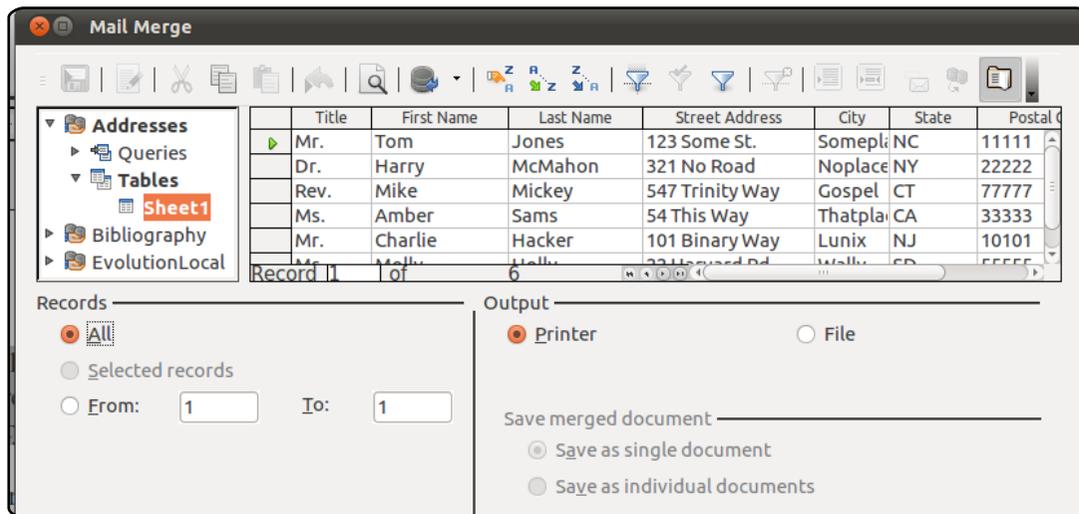
Title	First Name	Last Name	Street Address	City	State	Postal Code
Mr.	Tom	Jones	123 Some St.	Springfield	IL	62701
Dr.	Harry	McMahon	321 No Road	Springfield	IL	62701
Rev.	Mike	Mickey	547 Trinity Way	Springfield	IL	62701
Ms.	Amber	Sams	54 This Way	Springfield	IL	62701
Mr.	Charlie	Hacker	101 Binary Way	Springfield	IL	62701

<Title> <First Name> <Last Name>
<Street Address>
<City>, <State> <Postal Code>
Dear <Title> <Last Name>,
Start of letter..|

formatting toolbar. In the left part of the data sources, Addresses (or whatever you named your database) > Tables > Sheet1. On the right side, you will see your data laid out much like you entered it in the spreadsheet. We insert the fields into the document by clicking on the field header and dragging it into the document. This will create a field placeholder for every field you drag into the document. Make sure you drag the field header and not a data block

in a record.

Now, you can type the body of your letter. When you need a field from the database in your letter, you can drag it into the document. The database fields can be used any place in the document. As I said before, your database can contain whatever information you need it to contain. A teacher could even use it to report students' grades in a letter to parents.



Once you have finished typing your letter, you can print or send to a file. You might want to send a letter to file when you need to go back and add personal information for certain individuals. Like a teacher might want to request a parent-teacher conference for a student failing the class. File > Print, just like you would for a normal document. However, LibreOffice will tell you that your document contains address database fields and asks you if you want to print a form letter. Answer yes. The Mail Merge dialog will appear. In the dialog, you can choose to print to the printer or to a file. Also, you can choose to print a letter only for selected records or a range of records. You select records by clicking on the gray box

at the beginning of the record, or use CTRL-click to select only certain records, or SHIFT-click to select a series of records. When you print to a file, you can choose to print to one file, in which case each letter will begin on a new page, or save as individual files, in which case each letter should have its own file. You can select which database field to use for the file name. On my version of LibreOffice (3.3.3), I got one document whichever I chose. Apparently, they are still working on this feature.

You can also use your database to generate labels. File > New > Labels opens the labels dialog. Select your database and table. Move the fields over into the label

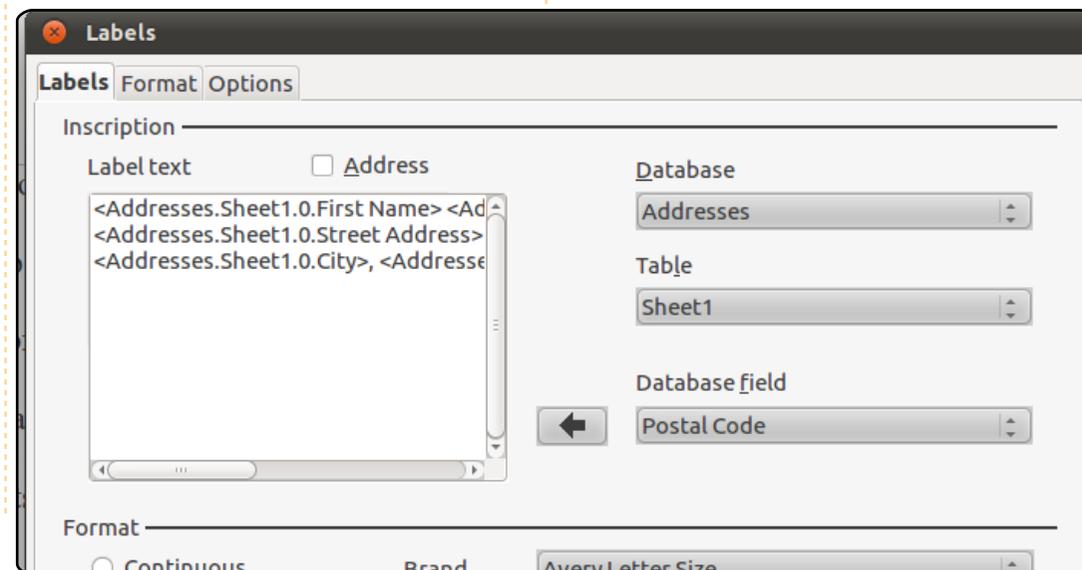
information box. Select your label type and click New Document. As with the form letters, when you get ready to print, you will have an opportunity to select the records you want to print, and you can print to a printer or a document. You can also create envelopes much in the same manner.

There is another way to create a form letter, using the Mail Merge Wizard: Tools > Mail Merge Wizard. The wizard has some restrictions, and when I tried to use the wizard, I found it more difficult to get good results. I found the manual method I have described here to give better results, giving the creator more control over the output.

Mail merge is a great time saver. Bang out your letter, drop in some database fields, and print. Sure beats typing 100s of letters or manually editing each one to change the information. In my next article, we will start to learn more about Calc by creating a simple budget spreadsheet.



Elmer Perry is a children's minister in Asheville, North Carolina whose hobbies include web design, programming, and writing. His website is eeperry.wordpress.com





This article will briefly explain the different files important to the packaging of Ubuntu packages which are contained in the `debian/` directory. The most important of them are `changelog`, `control`, `copyright`, and `rules`. These are required for all packages. A number of additional files in `debian/` may be used in order to customize and configure the behavior of the package. Some of these files are discussed in this article, but this is not meant to be a complete list.

The Changelog

This file is, as its name implies, a listing of the changes made in each version. It has a specific format that gives the package name, version, distribution, changes, and who made the changes at a given time. If you have a GPG key (see: Getting set up) make sure to use the same name and email address in `changelog` as you have in your key. The following is a template

changelog:

```
package (version)
distribution; urgency=urgency
```

- * change details
- more change details
- * even more change details

```
-- maintainer name <email
address>[two spaces] date
```

The format (especially of the date) is important. The date should be in RFC 5322 format, which can be obtained by using the command `date -R`. For convenience, the command `dch` may be used to edit `changelog`. It will update the date automatically. Minor bullet points are indicated by a dash "-", while major points use an asterisk "*". If you are packaging from scratch, `dch --create` (`dch` is in the `devscripts` package) will create a standard `debian/changelog` for you.

Here is a sample `changelog` file for `hello`:

```
hello (2.6-0ubuntu1) natty;
urgency=low
```

```
* New upstream release
with lots of bug fixes and
feature improvements.
```

```
-- Jane Doe
<packager@example.com> Thu,
21 Apr 2011 11:12:00 -0400
```

Notice that the version has a `-0ubuntu1` appended to it, this is the distro revision, used so that the packaging can be updated (to fix bugs for example) with new uploads within the same source release version.

Ubuntu and Debian have slightly different package versioning schemes to avoid conflicting packages with the same source version. If a Debian package has been changed in Ubuntu, it has `ubuntuX` (where `X` is the Ubuntu revision number) appended to the end of the Debian version. So, if the Debian `hello 2.6-1` package was changed by Ubuntu, the version string would be `2.6-1ubuntu1`. If a package for the application does not exist in Debian, then the Debian revision is `0` (e.g. `2.6-0ubuntu1`).

For further information, see the `changelog` section (Section 4.4) of the Debian Policy Manual.

The Control File

The control file contains the information that the package manager (such as `apt-get`, `synaptic`, and `adept`) uses, build-time dependencies, maintainer information, and much more.

For the Ubuntu `hello` package, the control file looks something like:

```
Source: hello
Section: devel
Priority: optional
Maintainer: Ubuntu
Developers <ubuntu-devel-
discuss@lists.ubuntu.com>
XSBC-Original-Maintainer:
Jane Doe
<packager@example.com>
Standards-Version: 3.9.1
Build-Depends: debhelper (>=
7)
Bzr-Vcs: lp:ubuntu/hello
Homepage:
http://www.gnu.org/software/h
ello/
```

```
Package: hello
```

Architecture: any
Depends: `$(shlibs:Depends)`
Description: The classic greeting, and a good example The GNU hello program produces a familiar, friendly greeting. It allows non-programmers to use a classic computer science tool which would otherwise be unavailable to them. **Seriously, though:** this is an example of how to do a Debian package. It is the Debian version of the GNU Project's 'hello world' program (which is itself an example for the GNU Project).

The first paragraph describes the source package - including the list of packages required to build the package from source in the Build-Depends field. It also contains some meta-information such as the maintainer's name, the version of Debian Policy that the package complies with, the location of the packaging version control repository, and the upstream home page.

Note that, in Ubuntu, we set the Maintainer field to a general address because anyone can change any package (this differs from Debian where changing packages is usually restricted to an individual or a team). Packages in

Ubuntu should generally have the Maintainer field set to Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>. If the Maintainer field is modified, the old value should be saved in the XSBC-Original-Maintainer field. This can be done automatically with the update-maintainer script available in the ubuntu-dev-tools package. For further information, see the Debian Maintainer Field spec on the Ubuntu wiki.

Each additional paragraph describes a binary package to be built.

For further information, see the control file section (Chapter 5) of the Debian Policy Manual.

The Copyright File

This file gives the copyright information for both the upstream source and the packaging. Ubuntu and Debian Policy (Section 12.5) require that each package installs a verbatim copy of its copyright and license information to `/usr/share/doc/${package_name}/copyright`.

Generally, copyright

information is found in the COPYING file in the program's source directory. This file should include such information as the names of the author and the packager, the URL from which the source came, a Copyright line with the year and copyright holder, and the text of the copyright itself. An example template would be:

```
Format:  
http://svn.debian.org/wsvn/de  
p/web/deps/dep5.mdwn?op=file&  
rev=166  
Upstream-Name: Hello  
Source:  
ftp://ftp.example.com/pub/gam  
es
```

```
Files: *  
Copyright: Copyright 1998  
John Doe <jdoe@example.com>  
License: GPL-2+  
This program is free  
software; you can  
redistribute it and/or  
modify it under the terms of  
the GNU General Public  
License as published by the  
Free Software Foundation;  
either version 2 of the  
License, or (at your option)  
any later version.
```

```
This program is distributed  
in the hope that it will be  
useful, but WITHOUT ANY  
WARRANTY; without even the  
implied warranty of  
MERCHANTABILITY or FITNESS  
FOR A PARTICULAR PURPOSE.
```

See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this package; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

On Debian systems, the full text of the GNU General Public License version 2 can be found in the file `/usr/share/common-licenses/GPL-2`.

```
Files: debian/*  
Copyright: Copyright 1998  
Jane Doe  
<packager@example.com>  
License: GPL-2+
```

This example follows the DEP-5: Machine-parseable debian/copyright proposal. You are encouraged to use this format as well.

The Rules File

The last file we need to look at is rules. This does all the work for creating our package. It is a Makefile with targets to compile and install the application, then create the .deb file from the installed files. It also has a target

to clean up all the build files so you end up with just a source package again.

Here is a simplified version of the rules file created by `dh_make` (which can be found in the `dh-make` package):

```
#!/usr/bin/make -f
# -*- makefile -*-

# Uncomment this to turn on
verbose mode.
#export DH_VERBOSE=1

%:
    dh $@
```

Let us go through this file in some detail. What this does is pass every build target that `debian/rules` is called with as an argument to `/usr/bin/dh`, which itself will call all the necessary `dh_*` commands.

`dh` runs a sequence of debhelper commands. The supported sequences correspond to the targets of a `debian/rules` file: “build”, “clean”, “install”, “binary-arch”, “binary-indep”, and “binary”. In order to see what commands are run in each target, run:

```
dh binary-arch --no-act
```

Commands in the `binary-indep` sequence are passed the “-i” option to ensure they work only on binary independent packages, and commands in the `binary-arch` sequences are passed the “-a” option to ensure they work only on architecture dependent packages.

Each debhelper command will record when it’s successfully run in `debian/package.debhelper.log`. (Which `dh_clean` deletes.) So `dh` can tell which commands have already been run, for which packages, and skip running those commands again. Each time `dh` is run, it examines the log, and finds the last logged command that is in the specified sequence. It then continues with the next command

in the sequence. The `--until`, `--before`, `--after`, and `--remaining` options can override this behavior.

If `debian/rules` contains a target with a name like `override_dh_command`, then when it gets to that command in the sequence, `dh` will run that target from the rules file, rather than running the actual command. The override target can then run the command with additional options, or run entirely different commands instead. (Note that to use this feature, you should Build-Depend on debhelper 7.0.50 or above.)

Have a look at `/usr/share/doc/debhelper/example` `s/` and `man dh` for more examples.

Also see the rules section (Section 4.9) of the Debian Policy Manual.

Additional Files

The Install File

The install file is used by `dh_install` to install files into the binary package. It has two standard use cases:

- To install files into your package that are not handled by the upstream build system.
- Splitting a single large source package into multiple binary packages.

In the first case, the install file should have one line per file installed, specifying both the file and the installation directory. For example, the following install file would install the script `foo` in the source package’s root directory to `usr/bin`, and a desktop file in the `debian` directory to `usr/share/applications`:

```
foo usr/bin
debian/bar.desktop
usr/share/applications
```

When a source package is producing multiple binary



packages, dh will install the files into debian/tmp rather than directly into debian/<package>. Files installed into debian/tmp can then be moved into separate binary packages using multiple \$package_name.install files. This is often done to split large amounts of architecture independent data out of architecture dependent packages and into Architecture: all packages. In this case, only the name of the files (or directories) to be installed are needed without the installation directory. For example, foo.install containing only the architecture dependent files might look like:

```
usr/bin/  
usr/lib/foo/*.so
```

While foo-common.install containing only the architecture independent file might look like:

```
/usr/share/doc/  
/usr/share/icons/  
/usr/share/foo/  
/usr/share/locale/
```

This would create two binary packages, foo and foo-common. Both would require their own paragraph in debian/control.

See man dh_install and the

install file section (Section 5.11) of the Debian New Maintainers' Guide for additional details.

The Watch File

The debian/watch file allows us to check automatically for new upstream versions using the tool uscan found in the devscripts package. The first line of the watch file must be the format version (3, at the time of this writing), while the following lines contain any URLs to parse. For example:

```
version=3  
http://ftp.gnu.org/gnu/hello/  
hello-(.*)tar.gz
```

Running uscan in the root source directory will now compare the upstream version number in debian/changelog with the latest available upstream version. If a new upstream version is found, it will be automatically downloaded. For example:

```
$ uscan  
hello: Newer version (2.7)  
available on remote site:  
http://ftp.gnu.org/gnu/hello/hello-2.7.tar.gz  
(local version is 2.6)  
hello: Successfully
```

downloaded updated package
hello-2.7.tar.gz
and symlinked
hello_2.7.orig.tar.gz to it

For further information, see man uscan and the watch file section (Section 4.11) of the Debian Policy Manual.

For a list of packages where the watch file reports they are not in sync with upstream, see Ubuntu External Health Status.

The Source/Format File

This file indicates the format of the source package. Currently, the package source format defaults to 1.0 if this file does not exist. You are encouraged to use the newer 3.0 source format. In this case, the file should contain a single line indicating the desired format:

- 3.0 (native) for Debian native packages (no upstream version) or
- 3.0 (quilt) for packages with a separate upstream tarball

If, for some reason, you wish to keep using the old format, please create this file and put 1.0 in it to be explicit about the source package version. This allows for the future removal of the 1.0

default for the package source format.

<http://wiki.debian.org/Projects/DebSrc3.0> summarizes information concerning, and the benefits of the switch to, the 3.0 source package formats.

See man dpkg-source and the source/format section (Section 5.21) of the Debian New Maintainers' Guide for additional details.

Additional Resources

In addition to the links to the Debian Policy Manual in each section above, the Debian New Maintainers' Guide has more detailed descriptions of each file. Chapter 4, "Required files under the debian directory" further discusses the control, changelog, copyright, and rules files. Chapter 5, "Other files under the debian directory" discusses additional files that may be used.



I started using Ubuntu in 2006, and made the complete switch in 2008. I have been employed in the environmental business since 1998, and started dabbling in database design using Microsoft Access 97 in 2000. I loved Access, and, at the time, I used it to manage clients' information for the company where I worked. I changed companies, and the Access database grew to manage cost estimates, basic employee info, invoicing, projects, mailing (electronic and regular), and there even was a module to make boring (drilling) logs. Later on, as a side job from my "home office," I sold databases to other companies.

Several times in my career, I thought about opening my own business. The ideas revolved around environmental consulting and database design. It seemed like a pretty good idea, since some of the databases I created back in 2002 are still kicking, and I had learned a good deal about environmental regulations, sampling, procedures and

whatnot. Not to mention business management (lots of hats are worn when you are in a small business). In this environment I also learned about software licensing.

We were using MS Office for our report writing, table (spreadsheet) creation, email, publications (brochures), and database management; MS Visio for our CAD needs; ARCVIEW for our geographical information software (GIS); Surfer for our isopleths (topographical-like contours); Peachtree for our accounting; Adobe Acrobat, and, of course, Norton antivirus. The tab to set up a computer with said software ran to a bit over \$3,000, not including the computer itself. ARCVIEW is a heavy lifter, and benefits from more than 2 GB of RAM, a powerful CPU, a dedicated GPU (midlevel at least), and a big monitor. Yearly software renewals were approximately \$500. I almost forgot, there was a Windows 2000 server sitting in a corner that housed our documents and managed the backups.

A small business has to spend a good amount of money to open a shop. Say what you will, but small businesses are what keep the country going. They are so important that the US government has an agency dedicated to small businesses, the U.S. Small Business Administration (<http://www.sba.gov>). Luckily for me, I have not had the need to set up my own shop, but I do some side-jobs - so the idea still hangs in the back of my head.

Ever since I started using Ubuntu, I looked at how can this "no cost" alternative be used to help small businesses. Furthermore, I had donated several refurbished computers to children without one, and to a church. I installed Ubuntu on them because the need was for Web surfing and office software - mainly with education purposes (finish assignments and create school reports), and there was little money available. These computers were not very different from what a small business would

use. With all the above in mind, I started to devise a list of software and hardware needed to run a small business. This is a "guide" that will come in parts. First, I will talk about hardware and OS, then office productivity and specific need software. The setup for this fictional environmental company will be as follows:

- Five employees. One of the employees will do accounting most of the time; two technicians who will mostly do field work and equipment maintenance, and will use a computer to enter project related data; and two "staff" who will do field work, report writing, project management, cost estimates, and run the daily administrative tasks.
- Two desktop computers, two workstations, two laptops
- One server
- One network multifunction printer and scanner
- Office productivity, accounting, CAD, and GIS software

For the hardware part, a small business would think of going to an electronics retailer or Web shop

and ordering the computers they want. Check the certified Ubuntu computers at <http://www.ubuntu.com/certification>. Ubuntu certification means that these computers will work out of the box with Ubuntu.

The easiest is to buy a computer with Ubuntu pre-installed. Dell, System 76, and ZaReason sell Ubuntu pre-installed computers. Although Dell is the largest of the three, it does not offer servers. Don't go crazy building systems with lots of RAMs and huge hard drives. You will be fine with 2 GB of RAM and 120 GB of disk space. In a business environment, data is not held on individual computers but on the server.

Although builds change over time, look at these models:

System 76
(<http://www.system76.com>):



- Wildebeest – it offers a dedicated GPU, remember to include the CD drive [D]
- Lemur or Gazelle [L]

- Elan [S]

ZaReason (<http://zareason.com>):



- Limbo – remember to include the CD drive and at least dual cores [D]
- Hoverboard, Strata or Verix [L]
- ZU 4110 [S]

Emperor Linux
(<http://www.emperorlinux.com>):



- All their laptops

[D] = desktop

[L] = laptop

[S] = server

Dell has only one build, so check the options. Dell servers can be bought with SUSELinux or RedHat, and as such I will not discuss them further. One thing to note is that when you buy a computer with Ubuntu pre-installed, it seems that you pay more than a similarly configured Windows computer. Why? Only the powers that be know. However, adding software will tilt the

balance in your favor. When considering laptops, think about Solid State Drives (SSD). According to Consumer Reports, laptops are more prone to repair than desktops. I can think of various reasons for this: cramped spaces that do not ventilate properly, and shock and vibrations from carrying them around and dropping them. SSDs make computers more expensive but your data is more secure.

The workstations have the same considerations as above - with the added requirement of a dedicated GPU, and, maybe, increased RAM. You can opt for a gaming GPU or a workstation GPU. Nvidia brands its workstation cards as Quadro or NVS (<http://www.nvidia.com/page/workstation.html>), and ATI brands its as FirePro (<http://www.amd.com/us/products/workstation/Pages/workstation.aspx>). The main boast of workstation GPUs is that they are certified by several CAD and GIS vendors. I use a Radeon 4670 with 1GB of RAM.

When buying the servers, consider buying support. You can view the different levels of support at

<http://www.canonical.com/enterprise-services/ubuntu-advantage/server>. Server setup can be difficult, and is different from desktop setup. Ubuntu server does not install a GUI, it is all terminal. If you are not comfortable using the terminal all the time, it can be daunting. Mind you, a GUI can be installed, but overall it is best to have the option of paid support for at least the first year if you are not completely comfortable. After all, the server is going to hold your precious data. You can check FCM Perfect Server Guide, or the Ubuntu server installation guide (<https://help.ubuntu.com/10.04/serverguide/C/index.html>) for information on how to set up your server.

Another build option is to buy all the parts yourself. You can get some discounts on buying in bulk. Check the components to make sure they will work with Ubuntu. I have used cheap MSI motherboards that I bought at CompUSA, and they have worked well. My Dell computer has a Foxconn motherboard, and most Intel motherboards are fully supported. Some components have longer warranties when

bought singly, but you will have to service the computers yourself. Another task to add to your list is to install the OS and additional software.

When you select the OS, you might be tempted to get the latest Ubuntu build. Consider it carefully. LTS editions have three-year support on the desktop and five-year support on the server. Regular editions have only eighteen-month support. LTS editions are issued every two years. What this means is that your upgrading strategy is well defined. You know that you will get updates for Canonical-backed software for that length of time. Afterwards, you are on your own.

A good strategy would be to start migrating to a LTS between the 24th and 30th month. The support team can help you with this.

Along with OS installation is codec support. You might think that you will just enable the restricted-extras packages and be done with it. In the US, and in many other countries in the world, it is illegal to use those codecs without buying a license. Since you are running a business, you cannot afford to be out of compliance, because it might cost you. Just read about Ernie Ball at http://news.cnet.com/2008-1082_3-5065859.html. Legal playback support is a must if you want to see and hear most content



in today's Web. Luckily for us, there is a Spanish company that sells such licensed codecs: Fluendo (<http://www.fluendo.com>). The full codec pack (<http://www.fluendo.com/shop/product/complete-set-of-playback-plugins/>) will set you back approximately \$40 US. I have them and they work great. Note that these codecs allow only decoding. You cannot legally encode wma, wmv, mpa or mp3 using these codecs. The same goes for DVD playback. If you want to have such playback, you can buy the Fluendo DVD player, or get PowerDVD for Linux, which is sold in the Ubuntu store. When I bought the Dell, CorelDVD for Linux was installed. Both Corel and PowerDVD are unusable in 10.04, that is why I bought the Fluendo DVD player. It works and it works well. Fluendo also has Moovida Pro, a multimedia manager and viewer that provides the codecs, the DVD player, an mp3 encoder and commercial DVD backup (you can copy commercial DVDs as iso images to your hard drive). The downside to it is that none of the codecs work outside Moovida Pro.

After you have set up all the computers and the server, think

about where you want to print your documents. Since I run a one-man shop, I use the HP Photosmart Premium (c309) shown below. Photo printing, scanning, duplex, and for those who need it, fax, make this a good unit. And HP provides more support for Linux than other printer manufacturers. When you plan to buy a HP printer, go to the HPLIP website at <http://hplipopensource.com> to determine what level of support your printer has. A small business will need more than what this printer provides. HP provides laser and inkjet multifunctions that will match your budget and need. One good example is the HP Color Laserjet CM2320fxi, (<http://h10010.www1.hp.com/wwp/c/us/en/sm/WF06b/18972-18972->



3328064-12004-3328083-3597338-3597361-3597470.html). It offers network connectivity, automatic duplex, scanning, a good amount of RAM, and an automatic feeder. The HPLIP website lists the printer as fully supported, which means that everything works. Xerox WorkCentre 6505/DN also looks like a promising machine (<http://www.office.xerox.com/multi-function-printer/color-multifunction/workcentre-6505/spec-enus.html>). The specs say that it will work with Linux and there is an RPM package to install the CUPS driver.

Now you only need to connect all your gear on a network. For that you will need a switch. (Or a router -- ed.) There are eight devices to connect plus your internet modem. If every connection is wired, that means nine ports. Netgear and Cisco offer a sixteen port switch at (<http://www.netgear.com/business/products/switches/smart-switches/GS716T.aspx>) and (http://www.cisco.com/cisco/web/solutions/small_business/products/routers_switches/200_series_switches/index.html). This will enable a very basic network as everything will be connected to this switch. You will need to set a static IP for

the printer and server - otherwise you will have trouble every time you power up the computer. For a more secure, and complicated setup, check with a local consultant, the forums, or the paid support that you may have acquired when you bought the server.

Right now you have all your hardware joined together. Next issue we will start talking about software needs.



Jesse is an industrial microbiologist from the environmental field who now works for the government. Along the way loving FOSS and ways to help people with it.

Below Zero

Zero Downtime



Below Zero is a Co-located Server Hosting specialist in the UK.

Uniquely we only provide rack space and bandwidth. This makes our service more reliable, more flexible, more focused and more competitively priced. We concentrate solely on the hosting of Co-located Servers and their associated systems, within Scotland's Data Centres.



At the heart of our networking infrastructure is state-of-the-art BGP4 routing that offers optimal data delivery and automatic multihomed failover between our outstanding providers. Customers may rest assured that we only use the highest quality of bandwidth; our policy is to pay more for the best of breed providers and because we buy in bulk this doesn't impact our extremely competitive pricing.



At Below Zero we help you to achieve Zero Downtime.

www.zerodowntime.co.uk



HOW-TO

Written by David O. Rowell

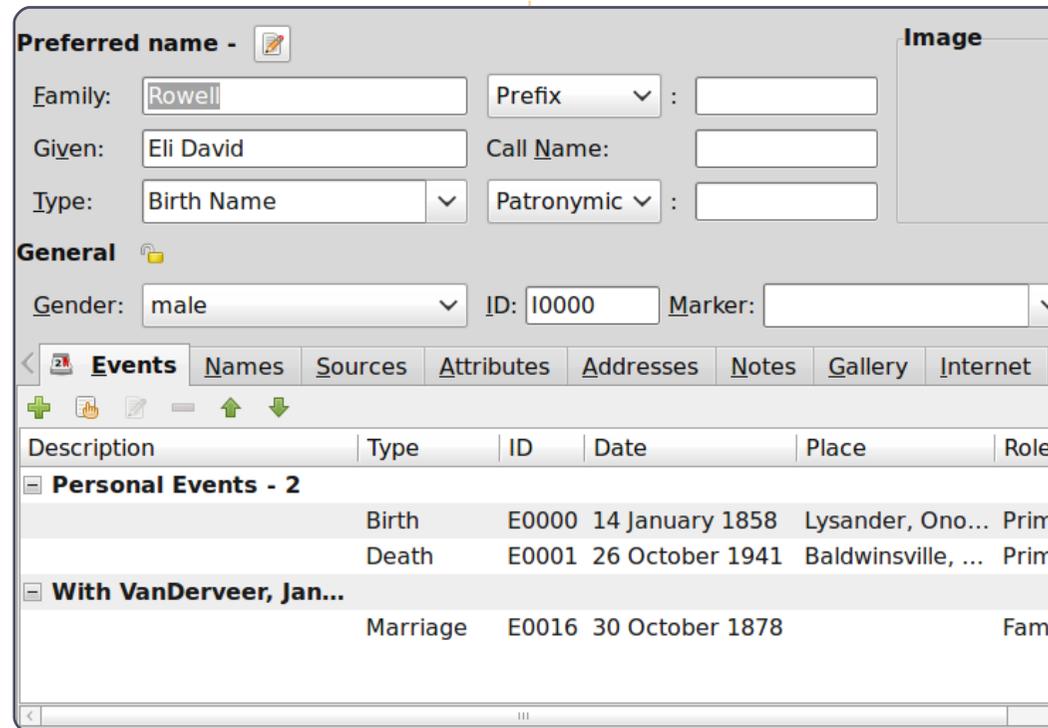
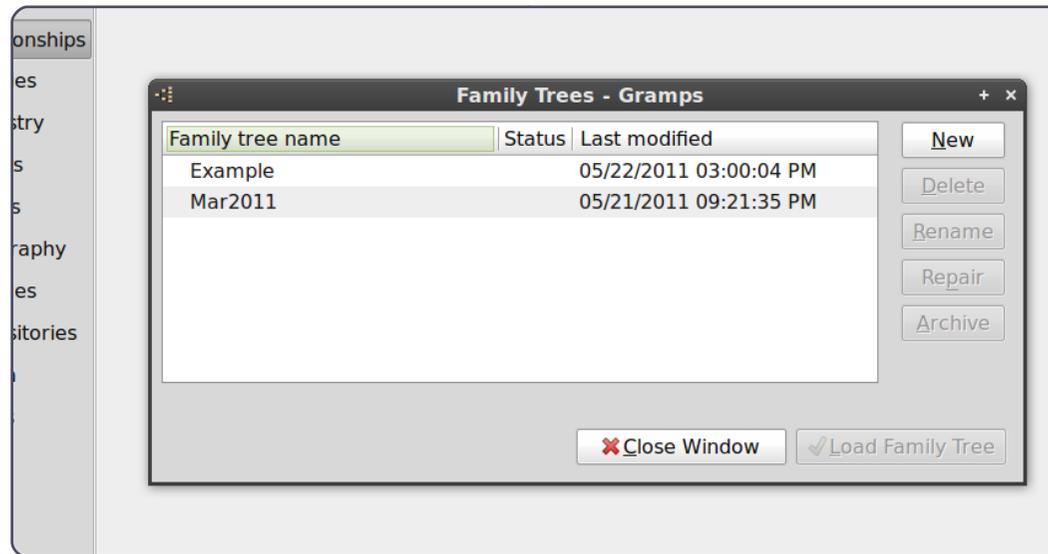
GRAMPS Pt.1

OK, you have an interest in genealogy and you've installed Gramps. Now let's use it. You'll find Gramps listed in the Menu's 'Office' category – click to start.

I've included some personal comments and observations on genealogy – I hope they will be helpful to beginners.

When you first start Gramps, you'll see the window shown to the right, except you won't have any databases listed. The program is capable of dealing with a number of separate databases. Here, it's asking which to use. Your obvious choice here is [New]. Give your new file a meaningful name reflecting its content, and press [Enter]. Note that it is generally a good idea to have a single database for your whole family rather than splitting into separate databases.

There are other ways to do this, but let's start by clicking the 'People' tab to the left, then + in



the toolbar. You'll be presented with a blank 'New Person' screen. **Note:** it is recommended practice to begin this journey by adding your own details as the first person, working backward in time from there – parents, grandparents... For obvious privacy reasons, I violated this policy. I'm not familiar with naming conventions very far outside of my WASP background – so, if you're having problems, the folks in the Gramps forum will be happy to help you.

- Enter your name and select your gender from the list – be sure to enter your gender here if the program doesn't guess it right. Fiddle with the various drop-down lists to see your choices.
- To enter the birth event, click the + under the 'Events' tab. For birth, it's obvious that the person was involved in a 'Primary' role. From the drop-down box, select 'Birth'. Enter the birth date. Chances are that you know your birth date, but there are choices that can be selected by clicking the calendar icon just to the right. You'll use them all some day! Dates are “usually” entered in day-month-

HOWTO - GRAMPS - PART 1

year format - but you don't have to. Just be sure that you're consistent. Click the + to the right of 'Place' to enter a new place name - you won't need to enter this place ever again. From the example, you'll get the idea of what and how to enter data. OK, from the top: Places are traditionally shown in small to large order - City, Township or Parish, County, State, Nation - as they existed when the event occurred. This is an important 'rule' - don't assume. Fill in the data under the 'Location' tab as appropriate. Except for 'Sources', I don't find the other tabs useful in most instances. Information you enter under 'Description' will appear to the left on the person screen.

• Now for the part you'll most

often ignore, and most often regret the oversight later - click the 'Sources' tab. Here you should enter something about where you found the information you just entered. Having a birth, death, or other primary information source, it's usually obvious what information should be supplied. In the example, Eli was born before records were kept in Lysander, but his father kept a diary which has been preserved. Note the 'Confidence' drop-down - it is important that you consider the source and evaluate its validity. Click the 'Repository' tab and enter exactly where you found the information. Whole books have been written on sourcing; some day you might want to read one. Meanwhile, a Google search will start you on the right path. The

information entered here should guide anyone (even you) to the actual source you looked at, and your evaluation of the information. I know it's boring and

New Place - Gramps

Place Name: Lysander, Onondaga, NY

Latitude: Longitude:

ID:

Location Alternate Locations Sources Notes Gallery Internet References

Street:

City: Lysander Township Church parish:

County: Onondaga State: NY

Country: US of A ZIP/Postal code:

Phone:

General Notes

Date:

Volume/Page:

Confidence: Normal

- **Shared source information**

General Notes Gallery Data Repositories References

Title: Major's Diary

Author: Major Edgar Rowell

ID:

Reference information

General Notes

Media Type: Manuscript

Call number:

- **Shared information**

General Notes Addresses Internet References

Name: Onondaga County Library - Local History

Type: Library

ID:

Note: Any changes in the shared repository information will be reflected in the repository itself, for all items that reference the repository.

HOWTO - GRAMPS - PART 1

eats into your valuable time but, it IS important to enter source information as you go.

- The 'Death' and 'Marriage' events are handled in much the same way. Note how the city, Baldwinsville, is listed first in the displayed death event.
- The 'OK' button closes screens – stores the information in the database – and returns to the previous screen.

You now have a base person to work from, and have a pretty good general idea of how to enter information. Click the 'Relationships' tab, to the left, then the 'Add a new set of parents' icon in the tool bar. The 'New – Family' screen will open a window (next page, top right) so that you can enter the base person's parent information. Clicking the + icon next to either parent will open the 'New-Person' screen – you already know how to handle that. Enter the parents' relationship type, date, and place information, while you're there. On the same screen you can enter the base person's siblings; the whole family from one screen. Don't forget the source information.

Select a person then click the

'Ancestry' tab. A traditional family tree opens (below right) with males tinted blue and females pink. Hover the mouse over any person and a small window will open showing even more information. Click a person and they become primary. The small arrows move the display forward and backward in time - generation by generation.

At this point, you've developed a good grasp of how Gramps

Father 🖱️ +

Name: _____

Birth: _____

Death: _____

Mother 🖱️ +

Name: _____

Birth: _____

Death: _____

Relationship Information

ID: Type: Marker:

Children Events Sources Atttributes Notes Gallery LDS

+ 🖱️ ✎️ - ⬆️ ⬇️

#	ID	Name	Gender	Paternal	Maternal	Birth Date

HOWTO - GRAMPS - PART 1

works. Let me introduce some of the other features:

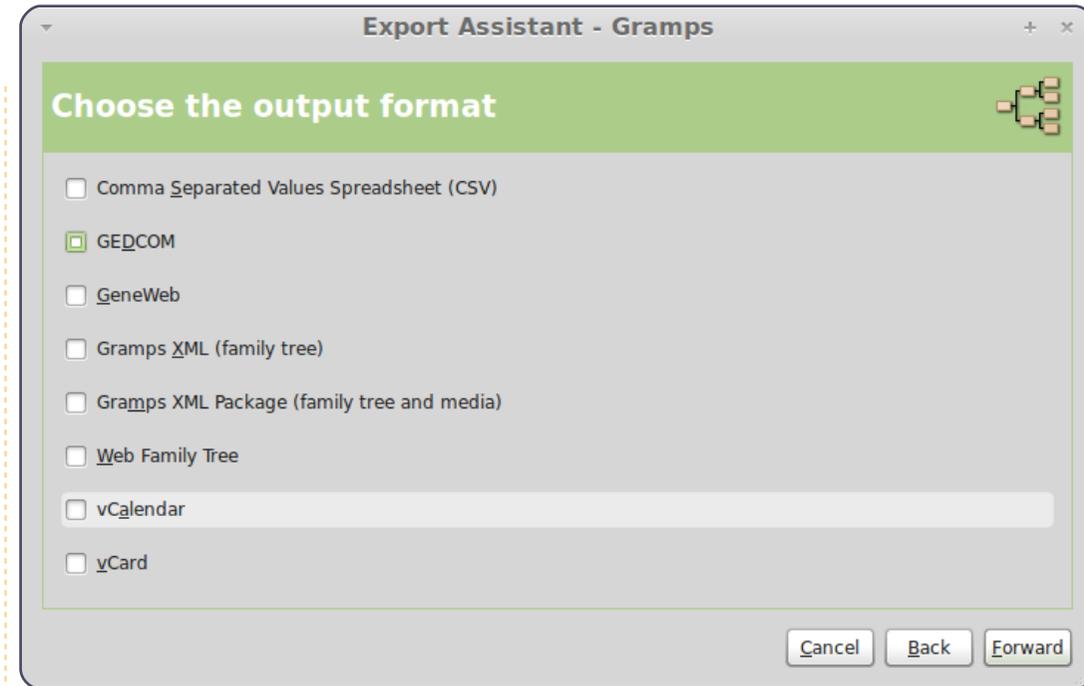
Here's a brief introduction into data import and export – in my experience Gramps plays reasonably well with other programs:

- To export the database, click 'Family Tree' > 'Export'. The assistant will open an informative window regarding export. Click 'Forward' and the Export Assistant's business window opens. Here you can choose from a number of export formats. For this example, we'll export the data in GEDCOM format – for my next article. Click the GEDCOM box to select it, then 'Forward'. Choose 'Entire Database' in both of the drop-down boxes, clear all of the remaining boxes, then click 'Forward'. Give the file to be exported a meaningful name, select the directory to contain the file, then click 'Forward'. A summary screen will open showing the options you've selected. If all is well, press 'Apply' to export the file. You should use one of the XML format options periodically to make a complete backup of your work.

- To import a GEDCOM, click 'Family Trees' > 'Manage Family Trees' > 'New', to create an empty

Gramps database. To import a file, click 'Family Trees', then 'Import'. Select the file to be imported, and click 'Import'. You may need to “correct” the imported file.

- A word about the GEDCOM file format. The LDS developed this file format to facilitate data transfer between genealogy programs. It's now in version 5.5, and is supported by any program worth serious consideration. BUT, you knew that was coming! All commercial programs, including Gramps, allow entry of data that isn't in the GEDCOM file definition. The result is that some data will be lost or garbled in the transfer process. Generally, a file of exceptions or problems will be displayed by the importing program – browse it. Until you're sure that the GEDCOM that you're dealing with is “clean”, DO NOT import into an existing database. Import into a new file, and carefully review the content before importing into your active database. I am disappointed with Gramps here. I exported the example database, and then imported it into a new file - there were several duplications and minor glitches that needed to be corrected. By-the-way, a GEDCOM is a simple text file that can be



edited with GEDIT. At times, a text editor is the simplest way to “correct” the file.

- In many of the screens you've been working with, there is a 'Gallery' tab. Working from there, you can add media objects (not restricted to pictures) to people, events, sources... It's useful to be able to link copies of birth, death, and marriage certificates to the event, but at other times a note will be a better choice. Copies of some sources are easy to link, and make a lot of sense - partial census images anyone? I recommend that you give serious thought to media organization before you get too

carried away here. I don't use this as my primary family history media display. My personal storage is organized under a main Genealogy directory. It contains many directories each with a family name prefaced with an Annentafel number (see text reports) to keep these directories ordered and at the beginning of the main directory. All media that I'm going to display in Gramps are dumped into a separate folder, chiefly so that Gramps can more readily backup the entire database. Another advantage to a separate directory is that images can be cropped and reduced without

HOWTO - GRAMPS - PART 1

affecting the originals.

- Most screens have a 'Notes' tab. I'd suggest taking full advantage of that to include source transcriptions, and bits of this and that, to more fully describe the person's life. Since a source can have notes, that may be a good place to document your evaluation.
- You have probably noted the 'Help' tab on the main program bar. Help is also just an [F1] away at any time. However you get there, the online Gramps Wiki Manual opens. The large table of contents will lead you to the information you need.
- That's all well and good, but how do we get information out of Gramps? Click the 'Reports' tab on the program bar, and a wealth of options is displayed. I suggest that you play around with all of them to find out just what they do and how to use them. Of course you could always consult the 'Help' function! I'd suggest that a printed 'Family Group Report' (FGR) be filed for reference as it contains a whole family's data in one place. An FGR in PDF format is a convenient way to send information to others.
- Gramplets are small plug-in applications that you may find useful in your work. Right-click the window background to see a list. I

suggest that you use the TODO Gramplet to list the tasks that you've found needful.

So where can you find online relevant information to fill out your family history?

- If you're a relative newcomer to genealogy, I'd suggest that you visit https://www.familysearch.org/learn/getting_started, and work through many of the guides provided. The Family Search site is provided by The Church of Jesus Christ of Latter-day Saints, the Mormons. I understand that family history is very important in their religion. LDS has undertaken some very large projects to digitize original records – much of this work is available for your research on the Family Search website at no cost. They also have a worldwide network of FamilySearch Centers. At these Centers, you can view microfilm of these records, and use some for-fee sites at no cost on their computers. Microfilm not found at a Center can be rented very reasonably from their main library in Salt Lake City. No, I'm not evangelizing here - it's just that LDS has generously provided these services for your use - at no cost, and with no strings attached.

- At some time, you will need to visit <http://www.rootsweb.ancestry.com/> to consult and register for some of their lists. You'll find a wealth of background information on RootsWeb. RootsWeb is hosted by Ancestry, but has remained free and independent of commercial influence.
- Other resources include www.worldgenweb.org, which will link to www.usgenweb.org, www.canadagenweb.org, and other national sites. The information you'll find here is organized geographically, and may contain information you need.
- If you can afford it, Ancestry.com has, probably, the most complete online digital content. You can search for names or places, find and view original and secondary sources, and

participate in online training. Considering the vast content, the price is reasonable.

- Google is your friend even for genealogy!

I hope that this has helped you to get started with researching your family history, and recording it with Gramps.





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).

Write your article in whichever software you choose. I would recommend OpenOffice, but **PLEASE SPELL AND GRAMMAR CHECK IT!**

Writing

In your article, please indicate where you would like a particular image to be placed. Please do not embed images into your Open Office document.

Images

Images should be JPG with low compression.

Regarding image sizes: if in doubt, send a full size screengrab and we will crop the image.

If you are writing a review, please follow the guidelines shown here.

For a more detailed list of the style rules and common pitfalls please refer to: <https://wiki.ubuntu.com/UbuntuMagazine/Style> - in short: US spelling, no l33t speak and no smilies.

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

If you can't write articles, but hang out in Ubuntu Forums, send us interesting forum threads that we could print.

Non-English Writers

If your native language is not English, don't worry. Write your article, and one of the proofreaders will read it for you and correct any grammatical or spelling errors. Not only are you helping the magazine and the community, but we'll help you with your English!

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.



Don't you just wish you could have your own CCTV system? It could help look after your car, your house, even your baby. Well, you can! ZoneMinder is a piece of software which - when hooked to a camera - will allow you to monitor, record, and even motion-detect for security, and all recorded to your PC. Not only that, but it can email you with a screenshot of the recording when a camera has detected motion.

In this example, I'm using my Logitech C310 HD webcam. It's automatically detected in most Ubuntu-based distros, and you can check that it's working by installing an application such as 'Cheese'. If the webcam is working in Cheese, then there's a good chance it will work in ZoneMinder.

So, with my webcam plugged in and working in Cheese, it's time to install ZoneMinder:

```
sudo apt-get install
zoneminder apache2 php5-
mysql libapache2-mod-php5
mysql-server ffmpeg
```

You'll notice that we're installing Apache and MySQL. Apache is installed because it's possible to control ZoneMinder via the Internet. The whole of ZoneMinder is controlled in a browser window. MySQL is needed as ZoneMinder saves most of its activations and such-like in a MySQL database.

Next, we need to get Apache talking nicely to ZoneMinder:

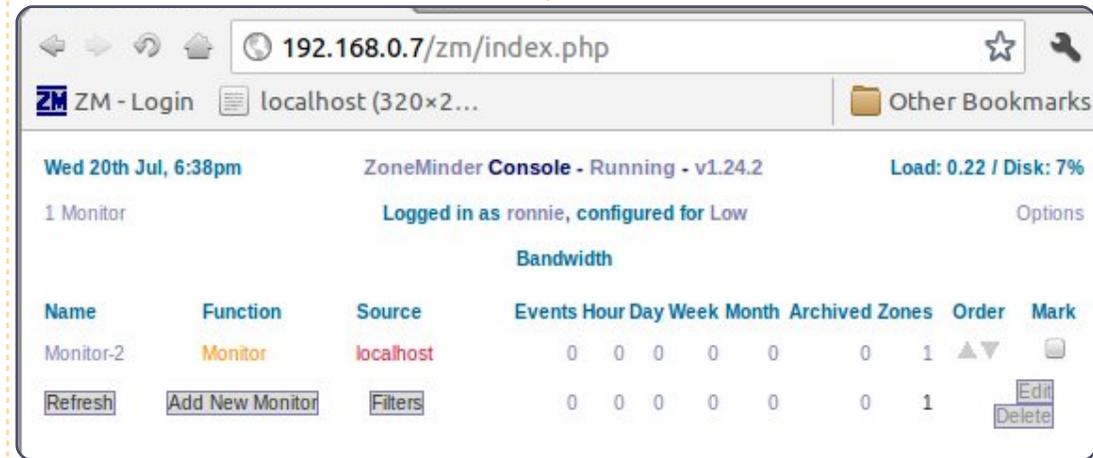
```
sudo ln -s
/etc/zm/apache.conf
/etc/apache2/conf.d/zoneminder.conf
```

With that done, we need to restart Apache with:

```
sudo apache2ctl restart
```

Now you should be able to get into ZoneMinder by going to your browser and pointing it to:

<http://localhost/zm/>



For reasons of security, I'd recommend that you create an admin account for yourself in 'Options' (under the 'User' tab), and, in the 'System' tab, turn on user authentication so that should anyone load up your browser/bookmark, they won't have admin access to your CCTV system!

This is where I hit my first major hurdle. My webcam is local. As in, it's sitting in the same room as me as I tinker with ZoneMinder. The other alternative is 'remote' which means the camera is not in the same room as me. No matter what I did, I could not get my camera to work with ZoneMinder. One major

flaw with ZoneMinder is that it doesn't just autodetect a camera and work, like Cheese does. The trick is to use mjpg-stream to stream the video from the camera to ZoneMinder. Thus, ZoneMinder now recognises it as a remote camera. This is when I hit my second major problem. The .deb package for mjpg-stream is no longer recognised as being valid (in the eyes of Ubuntu/apt-get), but I did manage to find a patched version which does work. So, you need to download MJPG from: <http://goo.gl/nFhNC>

Double click the downloaded file, or open it in Gdebi, and let it install. To run MJPG we open a

LINUX LAB - ZONEMINDER CCTV - PART 1

terminal and do:

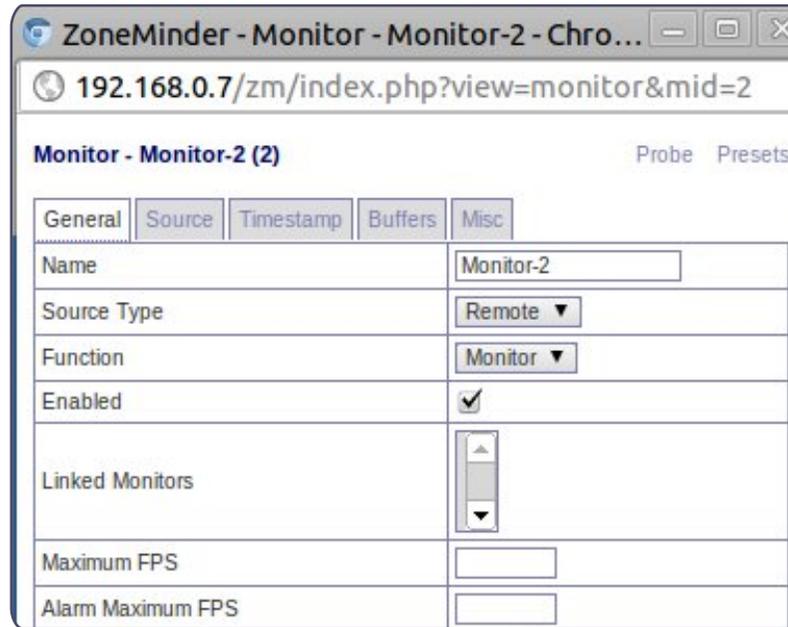
```
mjpg_streamer -i  
"input_uvc.so -r 320x240 -f  
6" -o "output_http.so -p  
8080" -b
```

What we're doing here is telling the camera to stream the video at 320 x 240 resolution, at six frames per second, via http to port 8080. Now we can configure ZoneMinder.

First click the 'Add New Monitor' button, and, in the pop up window, click the 'General' tab. From the drop-down menu beside 'Source Type,' choose 'Remote,' and, for the moment, we'll leave 'Function' as 'Monitor' - you can, if you wish, give your camera a meaningful name. This means we'll just be viewing the camera, not recording or doing any fancy motion-detection yet. Click the 'Source' tab, and in here we enter the screen width and height as 320 and 240 respectively, and the port as 8080. The hostname is localhost, and 'Remote Host Path' is (and don't forget the question mark at the start!):

`?action=stream`

And, finally, click the 'Save' button.



Before we can see anything in ZoneMinder, we need to change its default image-to-browser setting from JPG to MPG. This is done in the Options > 'Images' tab. Now we're ready to go.

Now, back in the ZoneMinder main screen, you should have your monitor (aka: camera) listed. Below the header of 'Name' should be your camera name (if you gave it a name) which you can click to see what your camera is streaming.

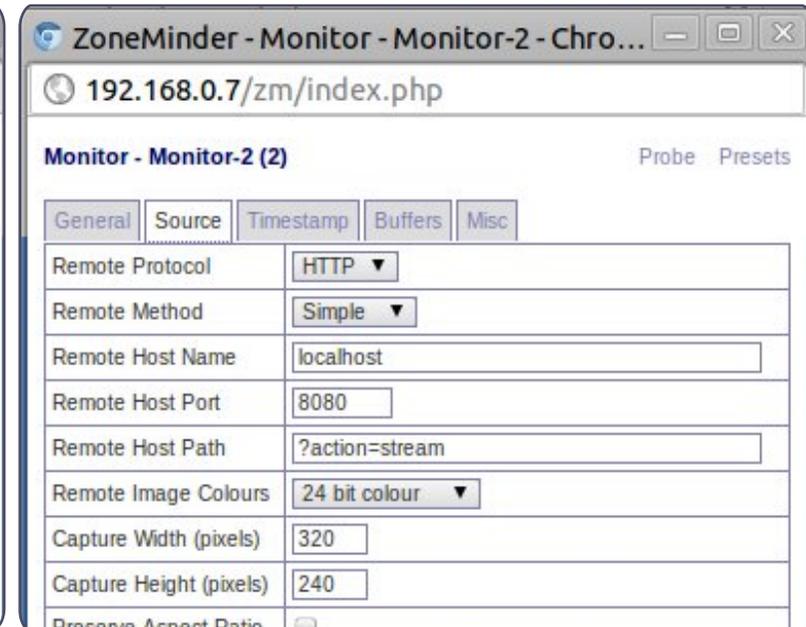
Now, the next problem. Our settings of 320x240 at 6fps is

hardly ideal, 640x480 at 30fps would be nice, but in some distros ZoneMinder has a shared memory problem which needs to be fixed before we can bump up the resolution.

To prove it's a ZoneMinder problem, we can test the camera output by going to the browser and entering:

<http://localhost:8080/?action=stream>

and you'll see your camera



LINUX LAB - ZONEMINDER CCTV - PART 1

at 320x240 at 6fps. Close the tab/window in your browser and kill MJPG from a terminal with:

```
killall mjpg_streamer
```

then enter:

```
mjpg_streamer -i  
"input_uvc.so -r 640x480 -f  
30" -o "output_http.so -p  
8080" -b
```

to set your camera at 640x480 at 30fps. Go back to your browser and do:

```
http://localhost:8080/?action=stream
```

and your camera will indeed be in 640 x 480 and displaying 30 frames per second.

Go back in to ZoneMinder and edit your camera settings to make the resolution 640 wide and 480 high, save it, go back, and check what your camera is streaming. You'll probably see nothing. If you see something then you're doing better than I did! If you see nothing, read on.

To fix things, we need to edit the sysctl.conf file by doing:

```
sudo gedit /etc/sysctl.conf
```

and below the line that says #kernel.printk we want to put:

```
kernel.shmall=268435456  
kernel.shmmax=268435456
```

and save it. You can close your text editor too; we shouldn't need it again. To use that new sysctl.conf file, we need to head over to the terminal again and do:

```
sudo sysctl -p
```

which will show you the stuff we just typed in. All good.

Get ZoneMinder in your browser again and you might notice that your source 'localhost' is in red. If it is, click it, then click 'Save'; that should fix it, and the text should go back to being orange. Click your camera (under 'Name') and it should work!

In the next part of this tutorial we'll enable some simple motion-detect capabilities, and also play back some recordings.

A PLEA ON BEHALF OF THE PODCAST PARTY

As you heard in episode #15 of the podcast, we're calling for opinion topics for that section of the show.

Instead of us having a rant about whatever strikes us, why not prompt us with a topic and watch for the mushroom clouds over the horizon! It's highly unlikely that the three of us will agree.

Or, an even more radical thought, send us an opinion by way of a contribution!

You can post comments and opinions on the podcast page at fullcirclemagazine.org, in our Ubuntu Forums section, or email podcast@fullcirclemagazine.org. You can also send us a comment by recording an audio clip of no more than 30 seconds and sending it to the same address. **Comments and audio may be edited for length. Please remember this is a family-friendly show.**

It would be great to have contributors come on the show and express an opinion in person.

Robin





MY STORY

Written by Jim Connett

One of the challenges facing non-profit organizations, specifically in the United States, involves how best to allocate resources donated to the organization in support of its mission. The economic turbulence felt around the world continues to test the resolve of those whose responsibility it is to continue offering services and support in the face of declining donations. Organizations desire to be on the cutting edge of technology, but also understand and embrace the challenges to operate up to a level acceptable to their donors and patrons.

In 2003, the people at the church I attend found themselves facing a major financial decision following the demise of their server, that ran Windows NT 3.51 for the last 10 years. The Board understood the financial ramifications of installing new Windows Server software, and the client access licenses (CALs) required to maintain current connectivity and productivity.

Fortunately, I seized an opportunity to suggest an alternative in using Ubuntu as a replacement. Our server requirements seemed simple enough; with my desktop experience on Ubuntu for a couple of years prior, I knew Ubuntu would support our needs well into the future. While we certainly lacked expertise in administering a Linux server, a couple of volunteers demonstrated a working knowledge of Ubuntu and the desire to learn more (including myself). After presenting the adverse financial impact of securing 20 CALs and the server OS from Microsoft, and then praising the power and capabilities of Ubuntu at a zero-cost basis, they expressed skepticism, but eventually agreed to it.

Over the course of a month, I worked to construct an Ubuntu server, and replicated all the shares from the old system under the Samba application, and even set up sizable home directories where users could back up private data from their desktop. After

overcoming a few obstacles in integrating both Mac and PC-based client operating systems, the Ubuntu server went live. It has been running ever since.

Through the use of WebMin, we administer this server through a GUI interface to accomplish all our needs, from adding users, modifying group membership, to adding and monitoring hardware. We even set up early warning systems through WebMin to alert us if file space becomes an issue, or if a particular service stops working. We've NEVER had a virus propagate through the server to any clients. While we experience our share of virus-infected clients (don't click on those email links!!), the virus's vector never involves the Ubuntu server. We even experienced a major power surge in 2010 that destroyed computer and networking systems, including the server containing our financial application, but Ubuntu quickly came back to life following the restoration of power to the facility. We aggressively use this server without the concern of

violating any end-user license agreements. We haven't even taken the time to upgrade to the latest "long term support" version! We're still running Ubuntu Hardy Heron 8.04! Someday, we'll upgrade, but the urgency is simply not there because we stay on top of our patches and updates for this version, and the system continues to run flawlessly. To say the installation and implementation of Ubuntu was (and still is) a smashing success is a complete understatement.

My hope is that anyone needing controlled network connectivity in any size organization would seriously consider giving Ubuntu a try. The learning curve for an administrator is very small due to the plethora of tools available, such as the aforementioned WebMin, and the many online forums outlining common problems and solutions. I am convinced Ubuntu is no longer a hobbyist operating system. It is a mainstream operating system providing server class capabilities at no cost to the organization.



MY OPINION

Written by Marcel Le Roux

Why is it that our secondary schools insist on buying the latest and expensive Microsoft Office software? Not only do they use our taxpayers' money, but they force the pupils, or at least their parents, to also buy the latest software so that they can complete their homework assignments at home. As a parent, this has been puzzling and frustrating me for the last 3 years.

"Why don't you use LibreOffice instead?", I hear you say.

As much as I love the open source alternatives, Microsoft seems to be doing everything in its power to make sure new formats, like docx and pptx, don't import into LibreOffice or OpenOffice without seriously messing up page layouts, text boxes, word art, fonts, text formatting and wrapping around pictures.

After wiping the slow Vista OS off my daughter's Dell laptop and replacing it with a much faster and more reliable dual boot Ubuntu

10.04 LTS and Linux Mint 11 system, she was constantly complaining that she had to waste time re-formatting all her documents and presentations when she switched between LibreOffice and the MS Office 2007 they use at her school. I've tried everything but just could not help her solve this annoying problem. Against all my instincts, I was just starting to consider buying the new Windows 7 OS to install on her laptop so that I could install Office 2007 seeing that I already had a disc with 3 licenses. It was then that I decided to search the web for help on installing Office 2007 using Wine, the Windows Compatibility Layer. After sifting through a number of conflicting posts, and trial and error, I managed to get it working, not on the latest Linux Mint 11 partition, but on the Ubuntu 10.04 partition instead. My biggest challenge was to activate the install because it simply would not let me type anything into the key activation box for some reason.

A Quick How-To

This is how you can do it: first, make sure you have the latest version of Wine installed by entering the following commands into the terminal:

```
sudo add-apt-repository  
ppa:ubuntu-wine/ppa
```

```
sudo apt-get update
```

```
sudo apt-get install wine1.3
```

In my case, I first used Synaptic to search for all "wine" related packages and marked them for complete removal. I also opened my home folder and set it to view hidden files and then deleted the .Wine folder. However, if you have any other software already installed under Wine, I would not recommend these last two steps.

Next, check in the Software Manager to see whether you have "PlayOnLinux" and "WineTricks" installed, and, if not, install both packages. You should then find PlayOnLinux under Games from the Applications menu. Now, run PlayOnLinux and click on "install". Wait for PlayOnLinux to update its list of supported Windows software packages, and click on "Office" from the list which appears on the left. You can now select from various versions of



Microsoft®
Office

Microsoft Office. I chose 2007.

It then asks you to insert the installation disc and point to the relevant folder - Office12 in my case. Now just follow the instructions on the screen to complete the installation. I chose the custom installation, and made sure I only had Word, Excel, PowerPoint, Publisher and shared tools selected. I understand MS Access will not work under Wine at all. Before you can use PowerPoint, you need to go to Applications > Wine > Configure Wine, and, on the "Libraries" tab,



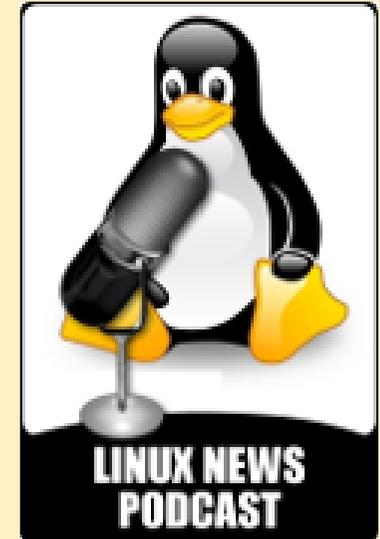
under "New override for library," type: "riched20", then click "Add". Now click the newly added "riched20" file, click "Edit", and select "Native (Windows)". Other posts suggest you follow the same procedure to add each of the following: "riched30", "msxml3", "msxml4", "msxml6", "corefonts", "tahoma", "vb6run", "vcrun6", and "msi2". Apart from a few suggesting additional font sets, I'm not sure what all the others are supposed to be doing. I entered all of them without breaking anything in Wine or Office.

If it all went well, you should be able to find the Office 2007 application icons under the Wine programmes menu from where you can send links to the desktop or the launcher panel.

I was very disappointed that the same procedure did not work on Mint 11 (based on the latest Ubuntu 11.04 Natty). However, under Ubuntu 10.04, Office 2007 has been working perfectly so far, apart from the fact that I'm unable to "send" any documents directly to email from within any of the Office applications. An easy workaround is to attach the documents to emails using

Evolution or ThunderBird.

I just wish that the government would embrace open source software - not only to save money, but also to encourage ICT teachers and their pupils to be more adventurous. Who knows, instead of just herding sheep, they may actually inspire the next Linus Torvalds, who could go on and invent something even better than Linux for the benefit of all mankind.



The podcast aims to be relevant by focusing on topic of interest to Linux users. We focus on such topics as software freedom, Linux development, Open Source software, Android and mobile devices, security issues, and Linux distribution releases. Special attention is given to the top ten Linux distributions: Ubuntu, Linux Mint, Fedora, Debian, openSUSE, Arch, PCLinuxOS, Puppy Linux, Sabayon, and CentOS.

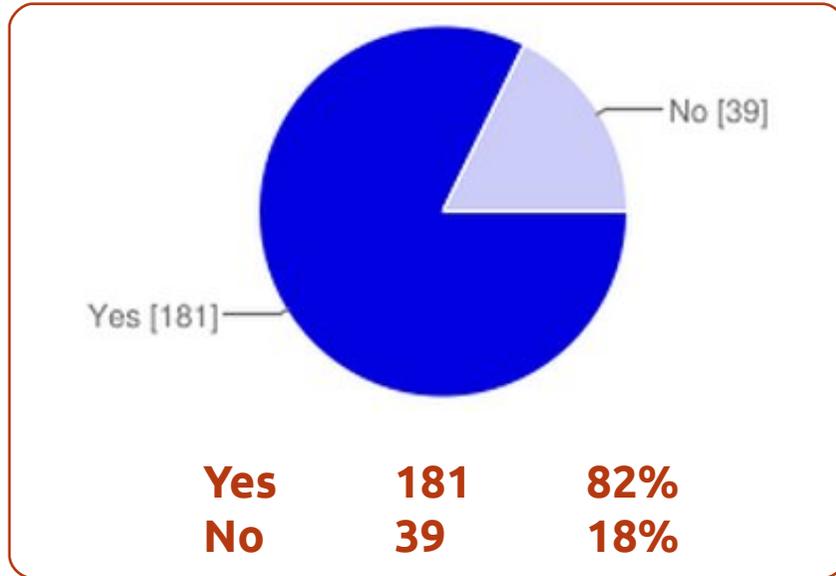
linuxnewspodcast.com



I THINK...

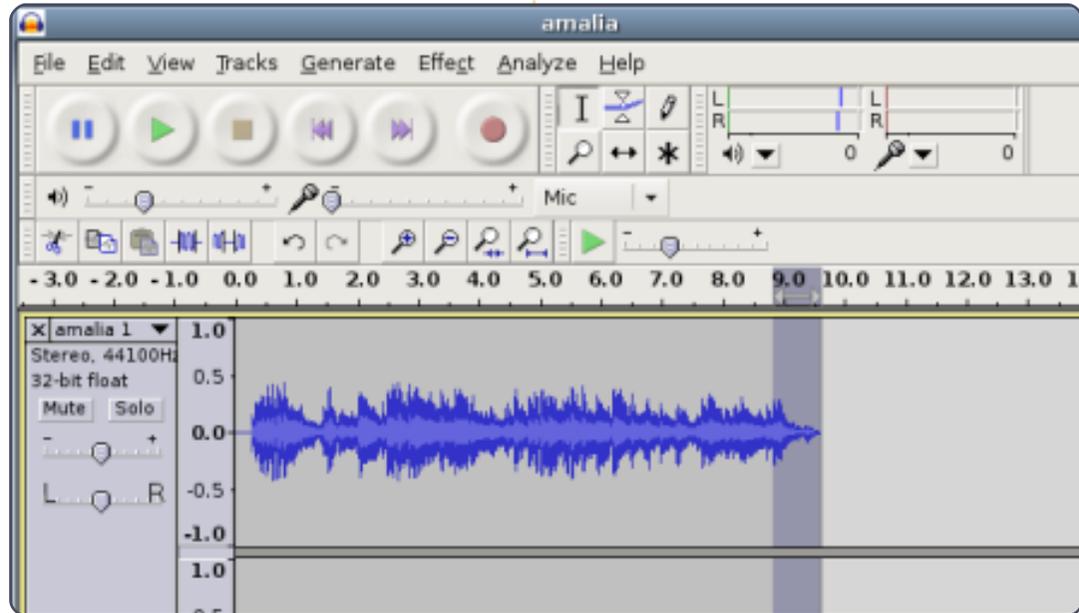
Last month's question was:

Would you like to see a series of articles on audio editing with Audacity?



- II Audacity is simple and well documented
- II Would be very interested since Audacity has lots of unknown functionalities for me!

II Sound editing is a topic which is not touched very often in magazines. It'd be awesome to have a series in FCM regarding casual audio editing in audacity.



- II Please include the swb plugins if you do the series.
- II Add narrative and music tracks to slide shows.
- II I need to know how to record anti-Unity comments in Audacity.
- II Please start with the basics!
- II It would be nice to see what Audacity can really do beyond the most basic editing

functions that I use.

- II I'm not interested in audio editing.
- II Yes. Noise removal & volume boosting without clipping!
- II I've just unsuccessfully tried to prepare a mix (simple fade-out, fade-in) but could not get the way to "align" the quasi-end of a song with the beginning of the next one.

The question I'd like to pose for FCM#53 is:

With the rise of web-based email, do we need an email client installed by default?

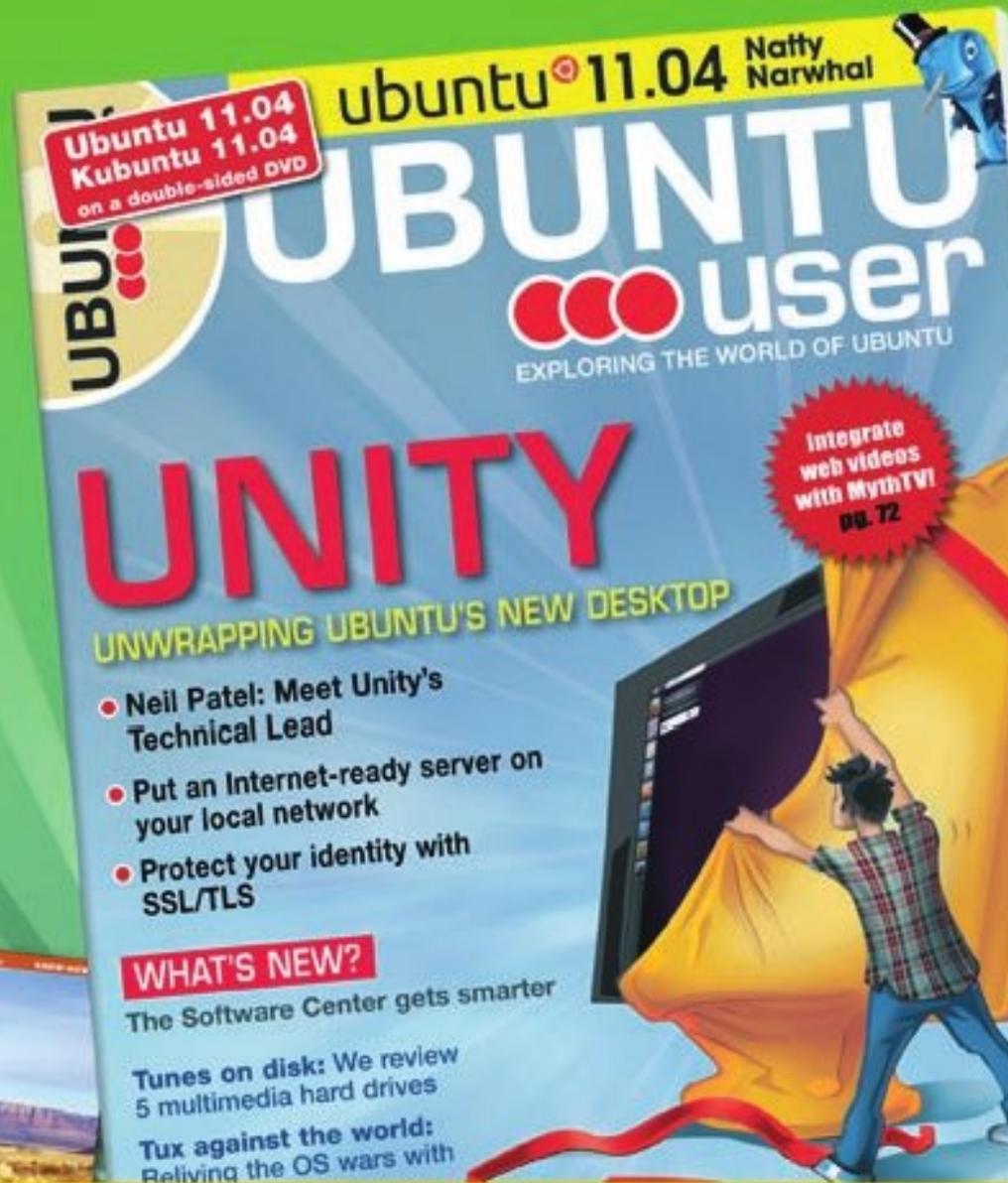
To give your answer, go to: <http://goo.gl/lF4Kd>

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



In 2008, Google wanted to speed up the Web and build a browser to support today's Web and the 'Web Apps' we all use each day. With the release of Chrome, we have seen Google build a super-fast browser which supports Web standards. With the help of Mozilla Firefox, we have seen the Web Apps become more feature-rich and faster, along with the decline of Internet Explorer usage, which has helped Microsoft finally realise that they need to build a better browser for today's Web.

Now, in 2011, we see Google taking the Chrome product further by releasing an operating system based around the principles of Chrome and the Chrome browser. With Chrome OS, Google wants to redesign the operating system to be fast, very secure, and simple to use, while providing a platform to support today's Web.

In this article, I will be reviewing both parts of the Chrome OS ecosystem - the operating system and the laptop powering the OS, aka a

Chromebook.

Chromebook

Before I review the Chromebook, I think it's best to explain what a Chromebook exactly is. At a very basic level, it is a laptop/notebook running Chrome OS. Some would argue that it is a netbook, but the Chromebooks currently on offer use displays larger than netbooks, full-size keyboards, and the hardware has a premium look/feel. Chromebooks do have some specific differences and requirements. There are no Function Keys (F1, F2, etc.); instead these are replaced with Chrome OS specific keys like 'Refresh', 'Back & Forward', and 'Full Screen'. There is no Caps Lock key, instead this is replaced with a Search button. The trackpad is very large, to support multi-touch gestures. All Chromebooks require an SSD to achieve their very long battery-life and fast boot.

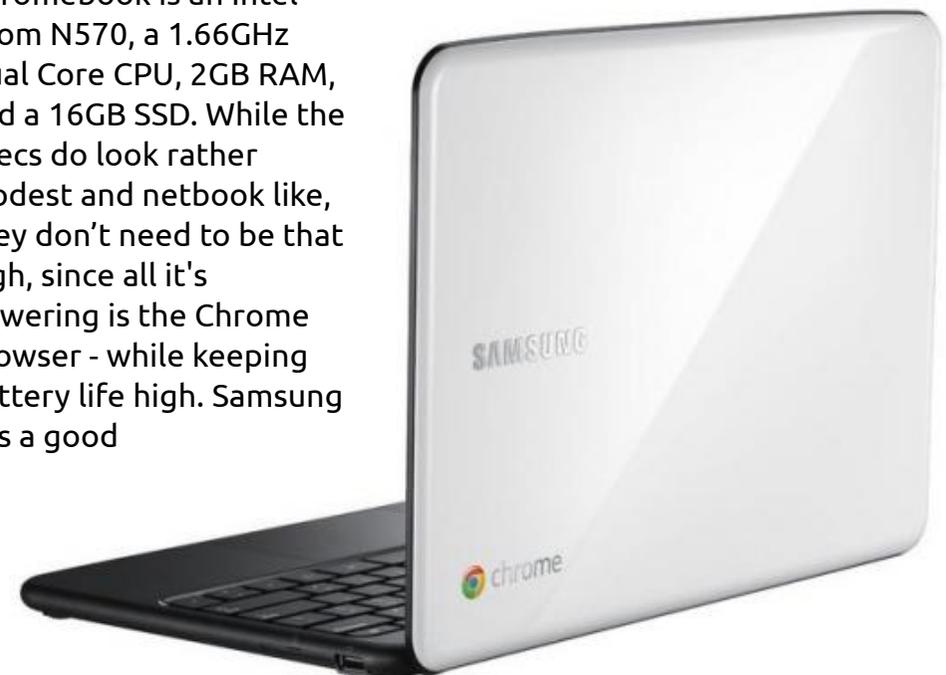
At Google IO, back in May, the Chrome team finally announced

the official release of Chrome OS as a commercial product, with a release date of June 15th. Chrome OS launched with two Chromebooks on offer, the Samsung Series 5 and Acer AC700. Both offer a WiFi-only model and WiFi + 3G model. I decided to buy the more expensive, premium, WiFi-only Chromebook from Samsung.

Inside the Samsung Chromebook is an Intel Atom N570, a 1.66GHz Dual Core CPU, 2GB RAM, and a 16GB SSD. While the specs do look rather modest and netbook like, they don't need to be that high, since all it's powering is the Chrome browser - while keeping battery life high. Samsung has a good

track record of producing excellent LCD displays, and they have not disappointed us with the screen on the Chromebook. It is a 12.1-inch super-bright LCD screen, with a resolution of 1280x800. Two USB ports, headphone port, multimedia card slot, HD Webcam, Mini VGA, and WiFi N finish the package.

The hardware is very solid and



has a premium feel. The screen is fantastic, and the resolution is a perfect mix of productivity and portability. The Island-style keyboard is a joy to type on - each keypress feels responsive. The large multi-touch trackpad is a great addition; I look forward to seeing multi-touch gestures included with Chrome OS.

After spending some time with the Samsung Series 5, I consider it to be very similar to the Apple MacBook Air. Similar size and weight, both trying to be lightweight ultra-portables. Both feature SSDs to improve performance and battery life. Both have similar size displays bundled with an HD webcam. Both feature an island-style keyboard and large multi-touch trackpad. The OS has been optimised and designed for the hardware. Both have an insanely high price tag. Both have sets of supporters who don't care how much it costs - they must have it to go with their iPhone/Nexus S. I know it's quite a comparison to make, but once you have seen both the Series 5 and Air, you will see the similarities.

I have not seen the Acer Chromebook yet, though, from

what I have heard, the specs are similar, but the hardware is not as high-quality as the Series 5. The battery life is only 6hrs, compared to Samsung's incredible 11hrs.

Chrome OS

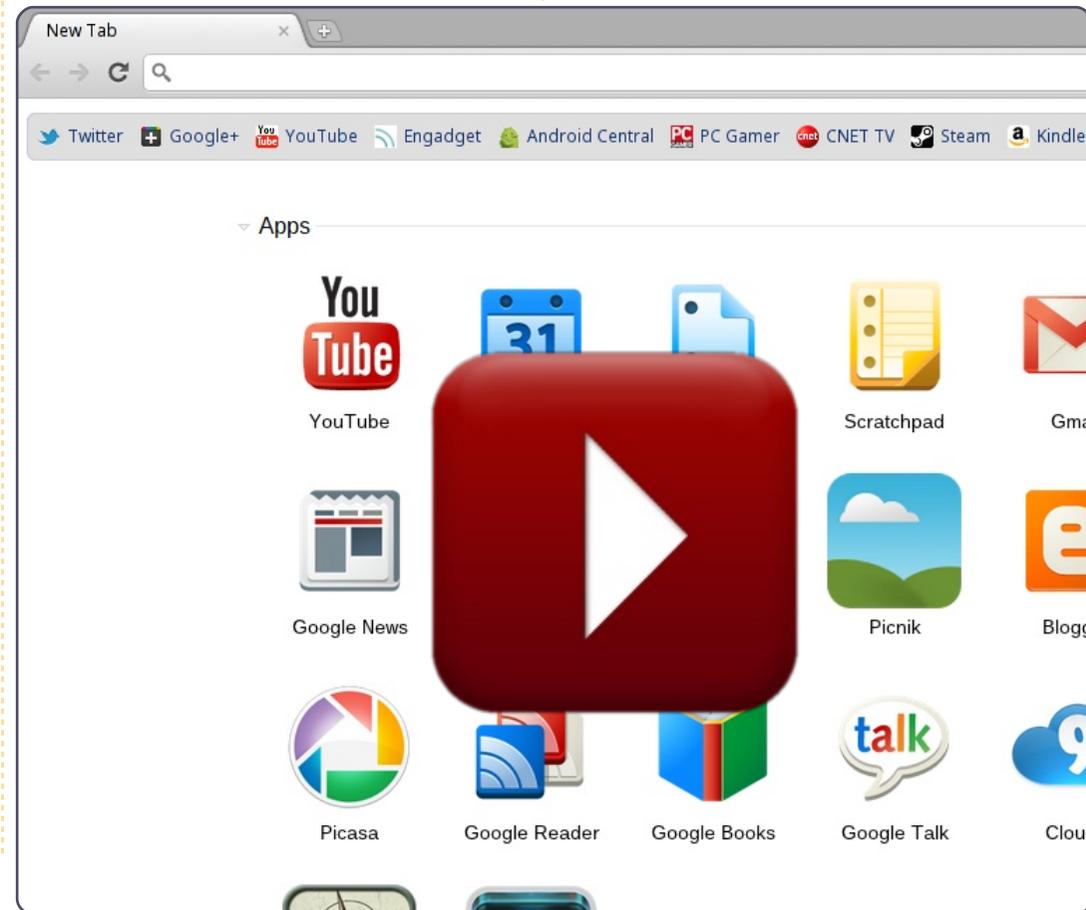
Chrome OS is a browser-only Linux operating system. It boots straight into the same Chrome browser we all love and know. That is it: nothing much more and nothing less - it is Chrome and the Web.

Chrome OS is powered by the Linux kernel, which is taken from Ubuntu, and will be upgraded twice a year, after each stable release of Ubuntu. The kernel has been optimised for the hardware Google knows the OS will run on, so that many drivers and boot checks can be removed to improve the performance and boot-time of the OS. Thanks to Google's efforts, Chrome OS does boot in 8 seconds or less, resumes from sleep instantly, and shuts down in a few seconds. By working closely with its hardware partners, Google has been able to optimise the drivers and kernel for performance and power management.

Security is another area which Google has focused on heavily. They believe they have created a security model which is very tough for hackers to breach and malware to attack. Plus, if such a rare attack happens, Chrome OS is able to detect this threat and remove it from its system. It achieves this very impressive security model by using the same technology it uses in the Chrome browser, by using a

multi-process architecture. Every Web page, plugin, add-on and OS, runs in its own process, and is protected from attack by a Sandbox. Google's own "Verified Boot" process is run during every boot up, which checks the system for malware, and, if a process is infected, a new version of Chrome OS is downloaded from the Internet.

After the 'Chrome' boot screen,



you're presented with the login screen, which allows you to login to your profile, which is linked to your Google account. At the login screen, you can also immediately login into a new account on your Chromebook. Or even log into "Guest Mode", which loads into Chrome in incognito mode.

Once you're logged in, you are presented with the Chrome browser. There is a brilliant experience when you login to your Chromebook for the first time and already have Chrome Sync enabled. All your Apps, Bookmarks, Settings, and Extensions, will immediately start appearing. For me, this was one of the main advantages to Chrome OS. Switch on your new Chromebook for the first time, and your system is up-to-date, all your settings and files are waiting for you within 1 minute of first boot.

The Chrome browser has a nice silver/gray look to it. I won't talk about the browser itself, you can all try it on your computer right now. It's the same experience - though there are some minor differences. In the top right-hand corner of the browser are system-specific indicators, such as the

clock, network, and battery. Notification and certain Web apps such as GTalk will all appear as individual windows which float above the browser.

All these windows are movable and can be minimized.

Chrome OS's window management is incredible, a far better experience than what

Windows, OSX & Ubuntu currently offer. Chrome OS supports virtual desktops; new windows of Chrome will display on different desktops, and can all be switched to very easily. I initially thought that virtual desktops in Chrome were pointless, but they make sense when you think back to how you manage your desktops in Ubuntu. You can have a set of tabs for social, set of tabs for work, set of tabs for entertainment, and so on.

One of the minor additions to Chrome OS compared to the Chrome browser is a built-in File Manager. This is used to manage your locally

stored files on your SD card and internal SSD. Any music, photos, and videos can be viewed on your

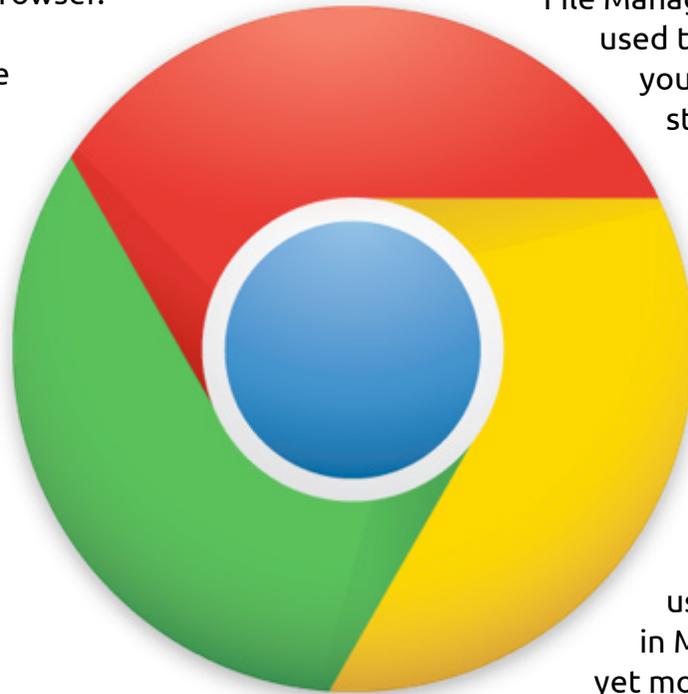
Chromebook using the built-in Media Player, yet more proof that

Chromebooks can be used offline. On the topic of offline support, currently only a few Web Apps work offline by using the HTML5 offline api. However, Gmail, Google Calendar, and Google Docs will have full read/write offline capabilities this summer.

Chrome OS's settings are very minimal, and can all be found in Chrome's standard settings tab.

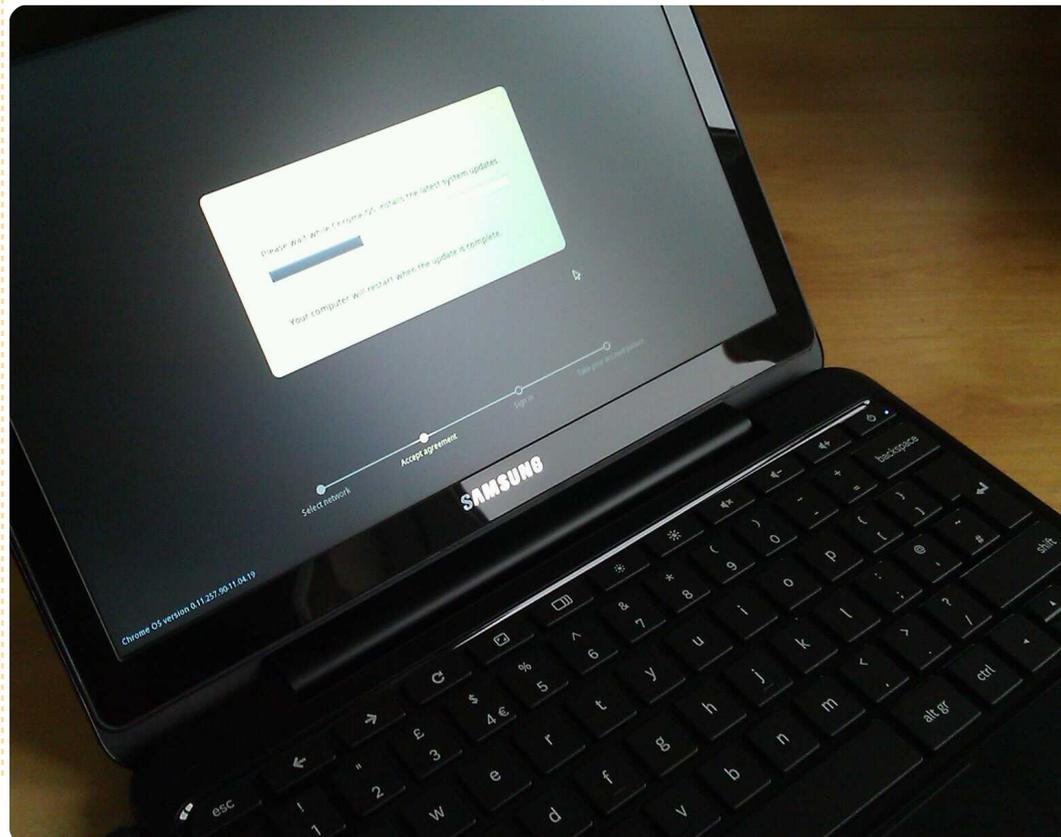
Examples are trackpad sensitivity, setting location/language, managing network connections, and even changing the Search button into a CAPS-Lock button again.

For all you power users out there, Google has allowed you to change the update channel between 'Stable', 'Beta', and 'Dev', allowing you to easily try out all the latest developments in Chrome OS. Google will be providing the same level of updates as it does to the Chrome browser; Chrome OS will be on the same 6-week update cycle. These updates will see the usual bug fixes, but also power/performance improvements and, of course, new features. As Google said back at Google IO, your Chromebook will update automatically and get better over time. I define the Samsung Chromebook as Google's "Nexus" Chromebook - for one main reason: the developer mode switch. Google has put a switch on the device which allows you to have full access to the kernel to start making improvements to the OS, and an unlocked boot loader so you can install other OSes on your Chromebook.



REVIEW: CHROMEBOOK

Overall, I believe Google has created an excellent desktop operating system which has been designed around the Web. You're on the Web in seconds, using the best and fastest browser currently out there. Chrome OS is very minimal, so you can fully utilise the website or Web app you're currently using. Chrome OS makes managing your Chromebook so easy. No need to worry about setting your device up, since all your settings and details are backed up to the cloud. No need to worry about viruses, or making sure you update your OS. All you need to worry about is just getting on the Web in seconds - this is what computers should be like. Stop worrying about the setup and maintenance; worry about getting your work done and just enjoy using your computer. I look forward to my Chromebook getting better over time, as I see improvements coming down the line to improve the Chrome browser and the underlying OS.



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



Any Takers?

Here's the situation:

- I have a desktop running Mint 11, a eeePC 701 running Crunch Bang, a Toshiba netbook running Windows 7 Cripple, a Kindle, an iPod Touch and lastly my wife has a big Toshiba notebook running Vista. The desktop is dual booting Windows 7 Home Premium. I'm still using Win 7 now and then and using its C:\ for primary storage.
- The desktop is connected to a switch/router using Ethernet, all the rest using WiFi. The router connects to our cable service. Internet access isn't a problem on any of them.
- Connected to the desktop is an HP Photosmart printer/scanner which I can use (at least as a printer) from the eeePC, the netbook and the big notebook so I assume I could from anywhere. THAT was an adventure!

• I also have a couple external hard drives connected to the desktop using Firewire. One is formatted fat 32 the other NTFS.

I'd like to share various folders on the computers with either everyone or, in a couple cases, just myself. I want to use one of the external drives as a dedicated back-up drive. I'm at a loss as to how to accomplish that which I think should be a relatively simple task. I just don't understand the gobbledygook surrounding networks. Back in the '90's I managed to set-up a couple peer-to-peer networks in the office but that ended with 'Windows for Workgroups'! You have to understand I'm 74 now and my tolerance for BS has deteriorated a lot!

David Rowell

Ronnie says: *Anyone out there able to write a HowTo on home networking for David, and the many others out there?*

LTS

Regarding Thomas's letter in FCM#51. Perhaps Thomas is new to Ubuntu so I would like to point him in the right direction. There is a version of Ubuntu especially for governments and large corporations which is exactly what he needs. It is called LTS (Long Term Support) and the versions which are LTS come out in the Spring in even years. The last one was 10.04 and the next one is 12.04 - these, therefore, are only changed every two years, and have a five year support cycle. Owing to the long support cycle, companies and governments can safely only change every four years.

If you are a hacker, and I use it in its true form, you have something to play with every six months, whereas if you are a business user or someone who hates change, you can keep it going for three to five years.

I use 10.04 LTS for work (I run a

Join us on:

-  facebook.com/fullcirclemagazine
-  twitter.com/#!/fullcirclemag
-  linkedin.com/company/full-circle-magazine
-  [ubuntuforums.org/forumdisplay.php?f=270](https://ubuntuforums.org/forum/display.php?f=270)

local newspaper) but at home I grab the first beta of every new release! At 72, I need a little excitement in life.

Andrew "Ampers" Taylor

GRAMPS

Iwanted to comment on the article about the GRAMPS genealogy program in the latest issue. I was very interested to see that there was a review of GRAMPS and will be looking forward to future articles about the software. It really is the

only way to go in Linux and was the final piece in the puzzle that allowed me to go Ubuntu full time. I wanted to make the switch but as a long time amateur genealogist I just had to be able to replace the Family Tree Maker software I had used for years. When I came across GRAMPS the problem was solved. It took some learning and the import of data from one program to the other wasn't as perfect as I had hoped, but it was a done deal once I got the hang of it.

I will be looking forward to seeing any future articles on this fine software.

Steve Barcomb

While the easiest data to find is online, if you get serious about genealogy you will find yourself poring over municipal and church archives, local to where your ancestors lived. The vast majority of this data has not yet found its way online.

John Cardiff

Wireless

My EEE PC Sheashell 1015PEM had a similar problem with wireless connection stability [FCM#49]. The solution was suggested somewhere on Ubuntu Forums - reinstall the wireless driver after each kernel update. Since Broadcom STA wireless driver package contains the source code - reinstalling it results in compiling the driver on getting a new kernel. This solved, for me, the disconnection problems. What I did is this:

1. Remove driver using System > Administration > Additional Drivers
2. Reinstall driver module. In my case it is command:

```
sudo apt-get --reinstall install bcmwl-kernel-source
```
3. Reboot the computer
4. Activate the driver using System > Administration > Additional Drivers

Artemy Vysotsky

KDE For Me

I didn't have the courage to go with Unity when Ubuntu 11.04 came out and instead opted for Kubuntu. In many ways I'm glad I did because I was introduced to some very good KDE applications and some new ways of working. Connecting to network drives so that LibreOffice can open files for example. There is much to like and I will probably carry on with Kubuntu because of its familiarity.

I was given another laptop at work and thought that I would see what all this Unity fuss was all about. I was really quite surprised once I jumped over a few hurdles, as much of what was agreeable about the good old Ubuntu was there under the hood. All one had to do was dig about a bit and things worked as expected. In

truth, I am rather pleased with Ubuntu 11.04. I have even come to like the launcher which I decided to emulate in Kubuntu. Perhaps the thing which I found hardest to remember is the menu drop downs being on the top bar of the screen. It takes a while for this behaviour to become natural. I am now using both variants of Ubuntu and find both have their merits. I have hopes that people do what KDE users came to realise. KDE 4.X is very good indeed and I am certain that Unity will also improve. Both desktops in my opinion are already good and will no doubt improve over time. Note that Windows has not been mentioned so far because although I am a Systems Admin in a Windows environment, Ubuntu allows me to do many things much more easily.

Iain Mckeand

kubuntu 



Hello! Today, I'm writing about diversity in free and open source software (FOSS). No, please don't run away! When I hear people talking about improving "the ratio" in FOSS communities, the issue usually is framed as an issue of social justice. We have a fundamental belief that FOSS ought to be egalitarian and open to all, yet it is clear that some types of people participate much, much more than others. Groups like Ubuntu Women, the Ada Initiative, and others, exist to determine and rectify the many causes of that inequality.

However, for many of us, discussions of diversity issues in FOSS (like the feminism "f-word" and other social justice issues in the world at large) have taken on a dreary, dismal air—invoking a gigantic, depressing, unsolvable mess that no individual could ever hope to fix. It seems every good FOSS conference these days has its Obligatory Diversity Talk where the speaker points at FOSS's dismal diversity statistics and

particular examples of bad behavior from over the last year, everyone leaves feeling guilty and despairing, and no one develops the energy to try to improve things.

There are FOSS projects, and user groups, with gender ratios far better than the average—for example, in 2009 a successful outreach program took the SF Ruby meetup group from two percent women to eighteen percent in a single year, and they continue to grow. A major key to practically addressing FOSS's diversity problem is that it isn't (just) about bringing more people belonging to group X into your project. It's about making projects proactively welcoming to outsiders and newcomers generally.

Let me be clear: sexist (and racist and otherwise biased) incidents, whether large or small, are bad, and they give FOSS a bad name. However, many of the impediments to more women (and other groups, though my research

is mostly on women in FOSS) getting involved in FOSS are far more subtle. These little obstacles, prerequisites, and annoyances accumulate to have their effect - molehills becoming mountains. They repeatedly filter the set of potential FOSS contributors such that nearly everyone who does manage to join the community comes from the extreme, rarefied end of the bell curve in several dimensions – not just gender or ethnicity, but also things like personality and skill set. Because these obstacles are individually small, though, they are highly solvable.

First, you're more likely to get involved in FOSS if you have a friend who's already involved, especially if he or she is willing to mentor you. For most FOSS projects, spread through friendship networks is practically the extent of any active "recruitment" whatsoever. Given FOSS's existing demographic profile, and the fact that usually our friends resemble ourselves, this sort of recruitment is unlikely



to bring in a more diverse contributor set.

Second, most FOSS projects are infamous for having poor documentation, not just for users but also for developers, and insufficient mentorship resources to make up for it. The lack of good docs suits people who have lots of free time (flailing around trying to get a working dev install running with crappy docs is very time consuming!); are highly confident in their technical prowess and problem-solving abilities (irrespective of their actual level

of skill or experience); and who have already contributed to other FOSS projects and are familiar with their tools and conventions. Women as a class have less leisure time than men, tend to be less confident in their technical abilities than equally-skilled men, and, as already stated, are less likely to already be involved in FOSS.

Finally, the usual FOSS project suits new contributors who are highly assertive, extremely persistent, and have a thick skin. Anyone who has enthusiastically submitted a patch to a new project, only to see it languish for weeks and find themselves practically begging an aloof project maintainer to review it, can speak of the frustrated, draining feelings that can be inflicted by a FOSS community unwelcoming of newcomers. Vicious flame wars, endless arguing over trivial features, and other inappropriate-yet-common behaviors in FOSS projects, are similarly discouraging to newcomers of any sort, but especially those who already feel like outsiders for demographic or other reasons.

Veteran contributors

sometimes tend to see FOSS' difficulty, obscurity, and antagonism as a sort of hazing ritual, with the belief that anyone worth keeping around in the community will soldier on despite them. Certainly, holding that belief is easier than the work it would take to clean up long-standing bad behavior in a given FOSS project. Either way, these characteristics turn away new contributors of all sorts, and especially tend to filter out women and other FOSS minorities.

The lifeblood of any major free software project is new contributors. New contributors provide fresh insight and energy, take on bugs and feature requests that old hands may have burnt out on, and increase your project's sustainability over time. It happens to be the case that most of the potential contributors out there—most of the people who could be making your project more awesome, but for various reasons aren't—probably don't look exactly like you. The flip side of women making up only five percent of Ubuntu contributors is that if that proportion were shifted—if women suddenly made up ten percent of contributors, or

the 20 to 30 percent ratio of the tech industry generally—that shift would represent hundreds or thousands more people, period, contributing to Ubuntu.

Focusing on under-represented groups within a FOSS project makes sense - not just as a matter of justice or a means of correcting implicit biases in the product and/or the community surrounding it. Under-represented groups in FOSS are your project's biggest recruitment opportunity. And, the great secret is: outreach and friendliness efforts don't just make your project more likely to attract more female contributors (or whatever demographic you had in mind); they put your project in a better position to recruit people who don't fit the usual FOSS contributor stereotype in any number of ways.

Think for a moment: if you were in charge of a Habitat for Humanity chapter, or a church Sunday School, or a university club, how would you try to recruit and integrate new volunteers? Normal humans like to be welcomed and shown how they can be useful. There's no reason FOSS projects can't do that for people.

Here are a few suggestions for making your project more welcoming: Trumpet the fact that you want new people. Provide mentors, resources, and special newbie-centric events to show new contributors the ropes. Keep a damper on negativity and hostility in your project's communication channels.

I am a contributor to the OpenHatch project, and we are one community you can turn to if you are wondering how to make those changes. We help projects run outreach events and highlight good tasks for newcomers, and those efforts have resulted in new contributors connecting with projects across FOSS. You can find us at <http://openhatch.org/>, and read about our events at <http://openhatch.org/wiki/Events>.

Whether or not you reach out to us, I hope that you try implementing ways to make your project more friendly and welcoming: the FOSS community's ability to grow both in absolute terms and in terms of diversity depends on it!



News

• **BEEP** released onto Linux - The adventure puzzle indie game, BEEP, has finally been released onto Linux. Currently only available on Gameolith for \$10, coming to Ubuntu Software Centre shortly.

Zombie themed games were a big hit during 2008 & 2009, with Valve releasing Left 4 Dead and other developers following suit with similar titles. Recently, we have seen interest again in the Zombie theme with the imminent release of Dead Island. Indie Developer, Blendo Games, has recently released a Zombie RTS title called Atom Zombie Smasher (AZS).

The story behind AZS is that the city of Neuvos Aires has been infected by Zombies, and your job is to save as many citizens as possible from this Zombie apocalypse.

The game's main feature is the single player campaign, in which you are presented with an overview map of Neuvos Aires, showing which areas have been infected by zombies. Each mission sees you launch an attack against the zombies in a certain area. The layout of the area is randomly generated for each mission, so replay value is very high here since you won't play the same mission-map twice. When launching an attack, you are given a detailed overview of the area of the city. You will see citizens as yellow/gold dots and zombies as pink/purple dots. First, you must set up a helicopter landing zone where you will pick up citizens. Defenses must be put in place to help secure the landing zone and reduce the zombie population. These defences range from snipers, bombs, artillery, and more. All these defenses can be upgraded throughout the campaign. Winning each mission requires you to meet a certain objective such as saving so many citizens or killing so many zombies.

The campaign is very enjoyable and easy to pick up. It has a very satisfying feeling, blowing up buildings and zombies! However, missions become difficult quickly, and can be repetitive over this 30+ mission campaign.

The campaign is the main bulk of the game; there are few extra features. Co-Op is one of the main features Blendo has been pushing, but it is sadly missing in the Linux version. A surprising addition is Mod Support, which is usually a

feature left for the major titles rather than Indie games. The wealth of mods available at the moment is rather good, helping to extend the replayability of AZS.

Blendo Games has done an excellent job providing after-launch updates to AZS. Many updates have been fixing and improving the game, especially adding new features and balancing.

The graphics and sound are



nothing to shout about, but do a decent job of providing an interesting 2D map of the city and the zombie invasion. The soundtrack is enjoyable and gives a very authentic 1950s vibe.

Atom Zombie Smasher is a very refreshing change for Indie games on Linux, providing a very enjoyable RTS game, which has good replayability. It does have its issues with missions becoming difficult quickly and repetitive, and it is very disappointing that Co-Op is missing from the Linux version. Solid graphics and gameplay, mod support, and updates from the developer, make this title a highlight for Linux Gaming.

It is a must buy for Linux gamers, and it is just excellent to see a RTS on Linux, a genre lacking on this platform. You can buy Atom Zombie Smasher from Blendo Games (<http://blendogames.com/atomzombiesmasher/>) and also download the demo.

As we go to (the virtual) press it has been announced that Atom Zombie Smasher can also be purchased through the Ubuntu

Software Centre.

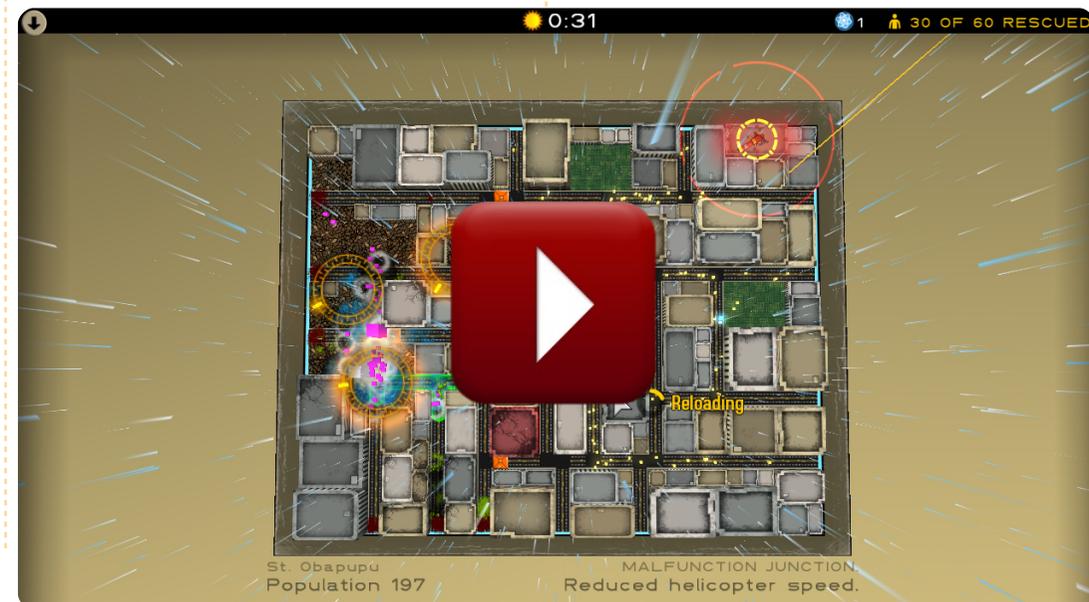
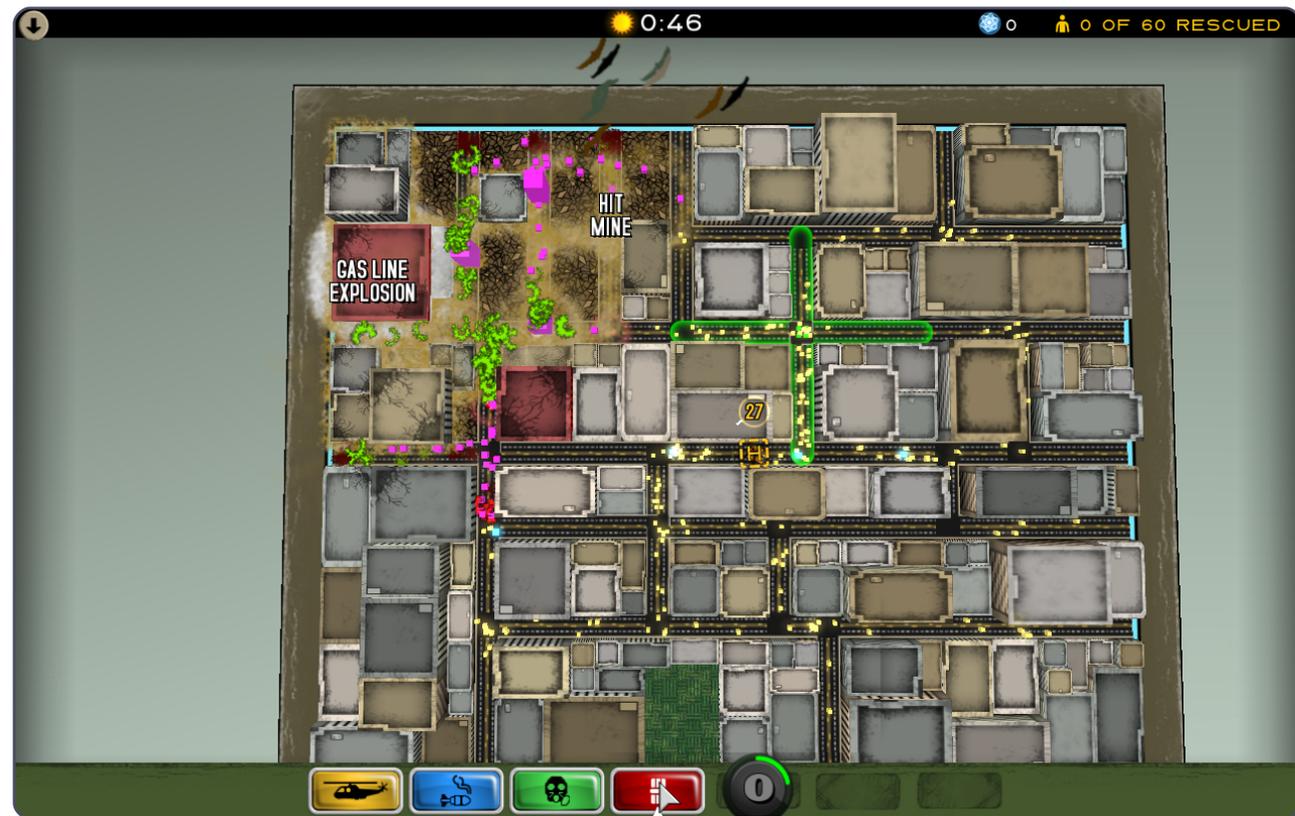
Score: 8/10

Good:

- Enjoyable Gameplay
- Unique Graphics & Sound
- Mod Support
- Plenty of Updates

Bad:

- Very difficult
- Missions become repetitive
- No Co-Op



Ed Hewitt, aka *chewit* (when playing games), is a keen PC gamer and sometimes enjoys console gaming. He is also co-host of the Full Circle Podcast!



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'm trying to fix a problem by modifying a file in the bin folder, but I get "permission denied."

A If you open Accessories/Terminal and enter the command:

```
gksudo nautilus
```

you can change the permissions. Be very, very careful, a wrong click can really mess up your system.

Q I want to remap the buttons on an xbox controller.

A Use xboxdrv. For information, go to: <http://linux.softpedia.com/get/Utilities/xboxdrv-37436.shtml>

Q I was decommissioning a machine and deleted some folders while doing so, while connected to Ubuntu One. When I synced my other machines, the folders were removed from them, as well.

A (Thanks to *duanedesign* in the Ubuntu Forums) Programs like DropBox and Ubuntu One are file sync services, they do not make good back up services. The file sync works both ways. Changes made locally are reflected on the cloud and vice versa. So if you remove a folder from the cloud, or your computer, it will be removed from all your computers.

If you need to recover a deleted folder please contact Ubuntu One support, and include the name of the folder, and we can try to recover that folder for you.

Gord says: *If you're going to decommission a machine, take it offline and re-partition the hard drive.*

Q My Dell Mini 10V has a 1.6 GHz dual core CPU. However, when I check, the CPUs are running at 800 MHz.

A (Thanks to 3rdalbum in the Ubuntu Forums) 1.6 GHz is the top speed of the processors. If they are virtually idle, they will drop back to 800MHz each to save power and reduce heat. If under load, they will shoot back to the full 1.6 GHz each.

Q I'm trying to set up Ubuntu Server 11.04 as a home file server, and I'm having some issues.

A It's much, much easier to install Desktop Ubuntu (perhaps the long-term release, 10.04), then use the Nautilus file manager to create and share (under the right-click menu) some folders.

"Server" is really meant for "headless" (no monitor, no keyboard) computers hosting high-volume web sites, or other applications where performance is an issue.

Q My Ubuntu is messed up so badly it will not boot. I have a LOT of files on the hard drive which I cannot afford to lose. How can I reinstall Ubuntu so I do not damage my files, and be able to pick back up where I left off?

A Hard drives fail! If you can't afford to lose the files, you must have backup. Go to your favorite computer store and buy two external hard drives which are each large enough to hold your data. Boot from a LiveCD, plug in an external hard drive, and copy your files onto it. Take the drive somewhere that it will not be destroyed if your home burns down. Plug in the other drive, and make another copy, which you can

Q & A

put in a safe place where you live. Once that's done, begin thinking about reinstalling Ubuntu.

Q I recently installed K9Copy and a lot of other packages were installed along with it.

Now I want to uninstall it, but only K9Copy uninstalls. How can I find and uninstall all the dependent packages that came along with it?

A (Thanks to *Chiel92* in the Ubuntu Forums) Open Accessories > Terminal and enter this command:

```
sudo apt-get autoremove
```

Q I have done a clean install of 11.04 but I'm having a little trouble finding drivers for my Broadcom BCM4306 802.11 b/g wireless.

A (Thanks to *IWantFroyo* in the Ubuntu Forums) You can go into Synaptic and search for 'b43'. There will be three firmware packages, each with a list of the cards they support in the description. Your card should be there somewhere.

Q I want to combine several MP3s into a single playable file.

A Run Synaptic Package Manager and install mp3wrap. It is a command-line program, so you use Accessories > Terminal to run it.

Q I am trying to share a USB external drive, formatted as NTFS, from my Ubuntu computer to my Windows systems.

A (Thanks to *Morbius1* in the Ubuntu Forums) Add a "force user" line to your smb.conf for the drive:

```
[HDMovies]
path = /media/HDMovies
comment = HD Movies 01
public = yes
only guest = yes
guest ok = yes
force user = jynks
```

After saving, restart Samba with this terminal command:

```
sudo service smbd restart
```

Q What's the deal with the new laptops with "switchable graphics," also known as "Nvidia Optimus"?

A See: <http://ubuntuforums.org/showthread.php?t=1657660> and <https://github.com/MrMEEE/bumblebee#readme>.

Q If I try to start my Lenovo G475 notebook without the adsl cable plugged in, it crashes!

A (Thanks to *drivard.com*) Turn on the Network Boot feature in the BIOS. (Holding F2 at boot up probably gets you into the BIOS setup on these computers.) This also affects some Acer notebooks.





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 screenshot, Ubuntu 10.04 with default wallpaper and theme. I was inspired by an Internet café in my hometown. I prefer using one panel on the bottom side and icon arrangement on screen. I've been doing some Ubuntu beginner lessons with neighbors, so I think with these settings they'll get familiar with it and it'll remain easy to understand.

My specs are:
AMD Athlon 64 x2 3600+ Dual-Core Processor, 2GB RAM, 80GB HD Drive, Nvidia 7300 GT

Budi Mulyono



The Conky is homemade with the batmanforeveralternate font. The lower dock is Cairo, with mostly Fekete icons - except for the launcher, which is a custom Awoken folder icon. The dock up right is AWN with Awoke icons. I also use cardpio for the menu.

All this, and much more, runs on my Belinea o.book 1301: Intel Centrino dual core 2GB RAM, and 13 inch screen which runs with a 1280x800 resolution.

K1au5



This is my Natty KDE desktop. Kwin can do all the things I know Compiz can do, but seems to handle crashes better. I am using a Wubi install within MS Vista. My desktop layout is set to "Search and Launch" which serves as a great replacement for Cairo Dock.

Coming from GNOME after 5 releases, KDE is finally as easy to use as GNOME and with better stability. However, I had to struggle to get Flash player to work for multimedia in Web browsing.

I have both Unity and KDE (no clashing I've come across).

My PC Specs:

Lenovo G530

Intel Dual Core 1.8GHz

3GB RAM

256MB Video RAM

Christo A Van Wyk



Here is my current Desktop.

PC Specs:

Ubuntu 11.04

Classic Desktop

RAM 2GB

Processor Intel Celeron 2GHz

Laptop: Inspiron 1525

Nothing too fancy over my computer specs; this is my computer at the office where I do most of my work. I use Awn (deleted all the GNOME panels), and Conky. Also covered my ALT+F2 needs with Synapse. Icons are Awoken Dark, and while I'm not doing presentations or out-of-office work, I switch over to Buuf icons.

Gerson Cordero



HOW TO CONTRIBUTE

We are always looking for new articles to include in Full Circle. For article guidelines, ideas, and for issue translation, please see our wiki:

<http://wiki.ubuntu.com/UbuntuMagazine>

Please email your articles to: articles@fullcirclemagazine.org

If you would like to submit **news**, email it to: news@fullcirclemagazine.org

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: www.fullcirclemagazine.org

FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your Opinions, Desktops and Stories. We also need Reviews (games, apps & hardware), How-To articles (on any K/X/Ubuntu subject) and any questions, or suggestions, you may have.

Send them to: articles@fullcirclemagazine.org

Full Circle Team



Editor - Ronnie Tucker

ronnie@fullcirclemagazine.org

Webmaster - Rob Kerfia

admin@fullcirclemagazine.org

Comms Mgr - Robert Clipsham

mrmonday@fullcirclemagazine.org

Podcast - Robin Catling

podcast@fullcirclemagazine.org

Editing & Proofreading

Mike Kennedy

Lucas Westermann

Gord Campbell

Robert Orsino

Our thanks go out to Canonical, the many translation teams around the world and to **Thorsten Wilms** for the current Full Circle logo.

Deadline for Issue #53:
Sunday 11th Sept. 2011.

Release date for issue #53:
Friday 30th Sept. 2011.

