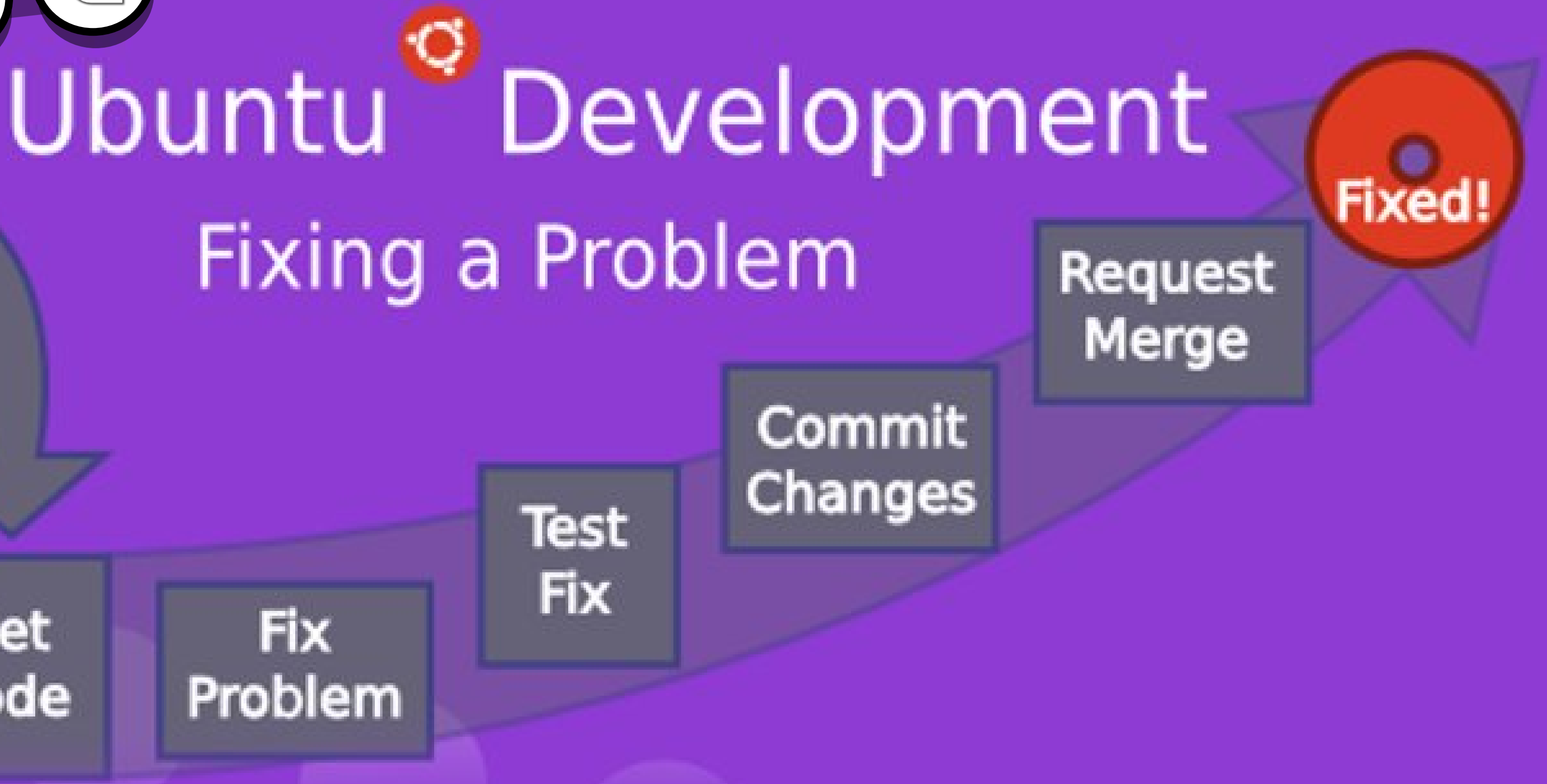
Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #51 - July 2011



HOW TO: USE KDE 4.6 - PART 2
DESKTOP EFFECTS AND APPLICATION EQUIVALENTS

Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

Welcome to another issue of Full Circle!

I have to say, I was very taken aback by all those requesting more KDE articles from me. I had assumed that KDE was still quite 'fringe', and not widely used. Seems I was very wrong. Even last month's question showed that KDE, while miles behind Gnome, is still quite popular, and that may increase as people take a dislike to Gnome-Shell.

For the second KDE article, I've focused on enabling desktop effects and listing some KDE equivalents to Gnome applications. Oh, and If you're wondering how to install KDE on your Ubuntu based distro, then you should check out the letters page. The Python and LibreOffice series continue, and the Ubuntu Development series reaches part three, where Daniel shows how to submit a bug fix.

If family history is more your thing, then have a look at this month's review of GRAMPS - the genealogy software. Starting next month, David Rowell will begin a series of articles showing how to use GRAMPS - beginning with creating a new database and entering names and details. So, keep an eye out for that.

My pile of My Desktop and My Story articles is getting quite low, so now is a good time to submit your desktop/story articles. Please include some info on how you got your desktop to look the way it does. But, don't let me stop you if you want to write an article on something else. All articles are welcome!

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

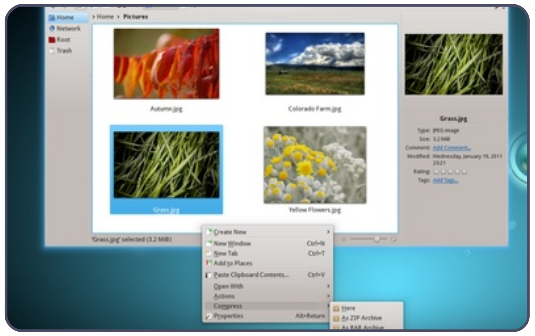
http://fullcirclemagazine.org

AUDIO MP3 AUDIO OGG

contents ^

LINUX NEWS

KDE 4.7 Released



KDE is delighted to announce its latest set of releases, providing major updates to the KDE Plasma Workspaces, KDE Applications, and the KDE Platform that provides the foundation for KDE software. Version 4.7 of these releases provide many new features and improved stability and performance.

• Plasma Workspaces Become More Portable
• Updated KDE Applications
• Improved Multimedia, Instant Messaging and Semantic Capabilities
• Instant Messaging integrated into desktop
• Stability As Well As Features

Source: KDE.org

Humble Indie Bundle #3

Humble Indie Bundle #3 has just been released. The games this time are: *Crayon Physics Deluxe, Cogs, VVVVVV, Hammerfight* and *And Yet It Moves*. As the website explains:

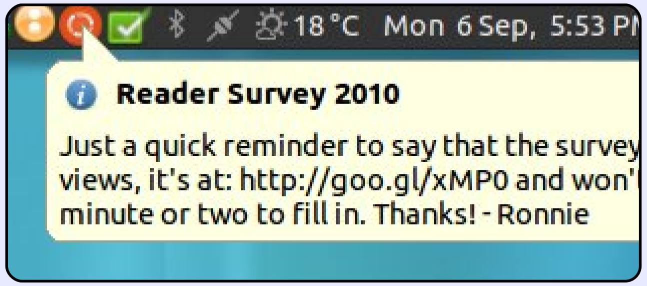
If you bought these five games separately, it would cost around $50, but we're letting you set the price! All of the games work great on Mac, Windows, and Linux.

As of writing the average Linux payment is $10.37. Average Mac payment is $5.43 with Windows at $3.47

Source: humblebundle.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**



18 °C Mon 6 Sep, 5:53 PI

Reader Survey 2010

Just a quick reminder to say that the survey views, it's at: http://goo.gl/xMP0 and won' minute or two to fill in. Thanks! - Ronnie

Indian Courts To Use Ubuntu

For the past four years all Indian Courts have been using RedHat Enterprise 5. Now, the Supreme Court of India has directed all Courts (approximately 17,000 of them) to change to Ubuntu 10.04. The Supreme Court of India has also given all Courts a customized Ubuntu DVD.

Each Court uses at least five computers. That's five computers multiplied by seventeen thousand Courts. That's 85,000 computers that will get Ubuntu

The Supreme Court of India committee page in which all the Indian Courts are directed to install Ubuntu is at: http://www.sci.nic.in/e-committee.htm

Source: A.Ramesh Babu (email)

contents ^

COMMAND & CONQUER

Written by Lucas Westermann

Dzen2 & Conky

Recently, I made the decision to move from WMFS (Window Manager From Scratch) to XMonad, since WMFS had started to present some issues when handling certain windows and layouts. Once I had made the switch, I was fighting with xmobar to get it working. Luckily, a guy on the ArchLinux Forum made the suggestion that I use Conky with dzen2 for my panel instead of Conky with xmobar, as I was trying to do. And so, I will be covering how to create your own status bar using dzen2 and Conky. Before I start I'd like to note that I am using a version of dzen2 that has xft support enabled. If you happen across a line in my configuration files/examples that is in the format "Togoshi Gothic:size=9", you'll need to replace it with a font from xfontsel, or else try the dzen2 packages from https://launchpad.net/~justinbogner/+archive/ppa/+packages, which seem to have XFT support. For those of you interested in my entire xmonad.hs, it's listed in the further reading section.

Below is my .conkyrc that I use for the status bar. I'll cover the important lines and explain what the scripts do. I won't be including my scripts, since they are either only for ArchLinux or are used for programs (like MPD and Dropbox) that not everyone uses. If you want a specific script, feel free to email me.

```
background no
out_to_console yes
out_to_x no
update_interval 2
total_run_times 0
use_spacer none

TEXT
${execi 1 /usr/bin/mpd-info}
| Dropbox: ${execi 5 echo
$(dropbox status)} |
$memperc% ($mem) | Updates:
${execi 300 python
~/Dropbox/Scripts/conky/packa
ges-short} | ${execi 60
python
~/Dropbox/Scripts/conky/gmail
.py} Email(s) |
^fg(\#9F6B00)${time %a %b %d
%H:%M}^fg()
```

The first line disables the background, and the next two disable the graphical aspect, so that Conky simply returns a string. The update_interval tells Conky how often to refresh the information. Total_run_times tells Conky to exit after a certain number of runs. Set it to 0 for infinite number of runs. User_spacer none tells it to not space out the commands below TEXT, since I do it by hand.

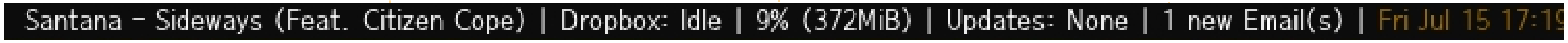
The following line of commands does the following:

```
<artist>-<song> | Dropbox:
<status> | % (<used RAM>) |
Updates: <# of updates> | #
new Email(s) | <clock>.
```

The clock is wrapped in ^fg(\#9F6B00)^fg(), so that dzen2 prints it in a nice gold colour, which matches my currently selected workspace (separate dzen2 instance). To see a screenshot, check the second link in the Further Reading section.

Once you've decided on your .conkyrc, you'll need to decide on the switches you want to use with dzen2. For that, you'll need to know the following switches:

-fg <hex> - sets the foreground colour using the hex value for the colour
-bg <hex> - sets the background colour using the hex value for the colour
-fn <font> - sets the font
-h <size in pixels> - sets the height
-y <y-coordinate> - shifts the bar up/down
-x <x-coordinate> - shifts the bar left/right
-w <pixels> - sets width of the bar
-sa <l,c,r> - set alignment of slave window
-ta <l,c,r> - set alignment of title windows
-xs <screen> - set the screen to display on.



```
Santana - Sideways (Feat. Citizen Cope) | Dropbox: Idle | 9% (372MiB) | Updates: None | 1 new Email(s) | Fri Jul 15 17:19
```

An example of how I call dzen2 for my workspaces (not the dzen2 instance with Conky):

```
dzen2 -fg '#9c9c9c' -bg
'#0c0c0c' -fn 'Togoshi
Gothic:size=9' -h 18 -y 0 -w
660 -ta l
```

An example of how I pipe Conky (it's a little more complicated the way I do it in my config file, but it's just easier to manage that way):

```
conky -c
~/.xmonad/.conkyrc_dwm_bar|dz
en2 -w 1040 -x 660 -ta r
```

The x-coordinate is the same as the width of the first bar, so that it lines up. You can also configure some default options for dzen2 using your .Xresources file in the format of:

```
dzen2.<property>: <setting>
```

Example:

```
dzen2.font: "Togoshi
Gothic:size=10"
```

Hopefully you've found this useful. For those of you who are going to use this to pretty up your Conky without using lua, or to those of you who run a window manager where there is no integrated status bar, I'd be interested to see how you guys put this information to use! If you have any questions, comments, or requests, you can reach me at lswest34@gmail.com. Please put “C&C” or “FCM” in the subject line of the email, so it doesn't get lost.

Further Reading:

http://pastebin.com/3g5TGQJJ – my xmonad.hs

http://lswest.deviantart.com/#/d3lalq7 - Screenshot

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

Which packages can I safely uninstall on Ubuntu Server ?
answer now
by Squeeze (445 points) in Ubuntu Linux - 0 votes
17 views
0 replies

How to access a Intel Express 535T Switch Hub
answer now
by GoldAlchemist (50 points) in Networking - 1 votes
34 views
4 replies

How to use wired desktop for server to wireless notebook
answer now
by txtinman (50 points) in Linux Servers - 1 votes
38 views
2 replies

Help with mod_security in Apache please

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

Written by Greg Walters

Program In Python - Part 25

A number of you have commented about the GUI programming articles and how much you've enjoyed them. In response to that, we will start taking a look at a different GUI toolkit called Tkinter. This is the “official” way to do GUI programming in Python. Tkinter has been around for a long time, and has gotten a pretty bad rap for looking “old fashioned”. This has changed recently, so I thought we'd fight that bad thought process.

PLEASE NOTE – All of the code presented here is for Python 2.x only. In an upcoming article, we'll discuss how to use tkinter in Python 3.x. If you MUST use Python 3.x, change the import statements to “from tkinter import *”.

A Little History And A Bit Of Background

Tkinter stands for “**Tk interface**”. Tk is a programming language all on its own, and the Tkinter module allows us to use the GUI functions there. There are a number of widgets that come natively with the Tkinter module. Some of them are Toplevel (main window) container, Buttons, Labels, Frames, Text Entry, CheckButtons, RadioButtons, Canvas, Multiline Text entry, and much more. There are also many modules that add functionallity on top of Tkinter. This month, we'll focus on four widgets. Toplevel (from here I'll basically refer to it as the root window), Frame, Labels, and Buttons. In the next article, we'll look at more widgets in more depth.

Basically, we have the Toplevel container widget which contains (holds) other widgets. This is the root or master window. Within this root window, we place the widgets we want to use within our program. Each widget, other than the Toplevel root widget container, has a parent. The parent doesn't have to be the root window. It can be a different widget. We'll explore that next month. For this month, everything will have a parent of the root window.

In order to place and display the child widgets, we have to use what's called “geometry management”. It's how things get put into the main root window. Most programmers use one of three types of geometry management, either Packer, Grid, or Place management. In my humble opinion, the Packer method is very clumsy. I'll let you dig into that on your own. The Place management method allows for extremely accurate placement of the widgets, but can be complicated. We'll discuss the Place method in a future article set. For this time, we'll concentrate on the Grid method.

Think of a spreadsheet. There are rows and columns. Columns are vertical, rows are horizontal. Here's a simple text representation of the cell addresses of a simple 5-column by 4-row grid (above right).

```
        COLUMNS - >
   ROWS  | 0,0 | 1,0 | 2,0 | 3,0 | 4,0 |
     |   | 0,1 | 1,1 | 2,1 | 3,1 | 4,1 |
         | 0,2 | 1,2 | 2,2 | 3,2 | 4,2 |
         | 0,3 | 1,3 | 2,3 | 3,3 | 4,3 |
```

So parent has the grid, the widgets go into the grid positions. At first glance, you might think that this is very limiting. However, widgets can span multiple grid positions in either the column direction, the row direction, or both.

Our First Example

Our first example is SUPER simple (only four lines), but shows a good bit.

```
from Tkinter import *

root = Tk()

button = Button(root, text =
"Hello FullCircle").grid()

root.mainloop()
```

Now, what's going on here? Line one imports the Tkinter library. Next, we instantiate the Tk object using root. (Tk is part of Tkinter). Here's line three.

```
button = Button(root, text =
"Hello FullCircle").grid()
```

We create a button called button, set its parent to the root window, set its text to “Hello FullCircle,” and set it into the grid. Finally, we call the window's main loop. Very simple from our perspective, but there's a lot that goes on behind the scenes. Thankfully, we don't need to understand what that is at this time.

Run the program and let's see what happens. On my machine the main window shows up at the lower left of the screen. It might show up somewhere else on yours. Clicking the button doesn't do anything. Let's fix that in our next example.

Our Second Example

This time, we'll create a class called App. This will be the class that actually holds our window. Let's get started.

```
from Tkinter import *
```

```
class App:
    def __init__(self, master):
        frame = Frame(master)
        self.lblText = Label(frame, text = "This is a label widget")
        self.btnQuit = Button(frame, text="Quit", fg="red", command=frame.quit)
        self.btnHello = Button(frame, text="Hello", command=self.SaySomething)
        frame.grid(column = 0, row = 0)
        self.lblText.grid(column = 0, row = 0, columnspan = 2)
        self.btnHello.grid(column = 0, row = 1)
        self.btnQuit.grid(column = 1, row = 1)
```

This is the import statement for the Tkinter library.

We define our class, and, in the __init__ routine, we set up our widgets and place them into the grid.

The first line in the __init__ routine creates a frame that will be the parent of all of our other widgets. The parent of the frame is the root window (Toplevel widget). Next we define a label, and two buttons. Let's look at the label creation line.

```
self.lblText = Label(frame,
text = "This is a label
widget")
```

We create the label widget and call it self.lblText. That's inherited from the Label widget object. We set its parent (frame), and set the text that we want it to display (text = “this is a label widget”). It's that simple. Of course we can do much more than that, but for now that's all we need. Next we set up the two Buttons we will use:

```
self.btnQuit = Button(frame,
text="Quit", fg="red",
command=frame.quit)

self.btnHello =
Button(frame, text="Hello",
command=self.SaySomething)
```

We name the widgets, set their parent (frame), and set the text we want them to show. Now btnQuit has an attribute marked fg which we set to “red”. You might have guessed this sets the foreground color or text color to the color red. The last attribute is to set the callback command we want to use when the user clicks the button. In the case of btnQuit, it's frame.quit, which ends the program. This is a built in function, so we don't need to actually create it. In the case of btnHello, it's a routine called self.SaySomething. This we have to create, but we have a bit more to go through first.

We need to put our widgets into the grid. Here's the lines again:

```
frame.grid(column = 0, row =
0)

self.lblText.grid(column =
0, row = 0, columnspan = 2)

self.btnHello.grid(column =
0, row = 1)

self.btnQuit.grid(column =
1, row = 1)
```

First, we assign a grid to the frame. Next, we set the grid attribute of each widget to where we want the widget to go. Notice the columnspan line for the label (self.lblText). This says that we want the label to span across two grid columns. Since we have only two columns, that's the entire width of the application. Now we can create our callback function:

```
def SaySomething(self):

   print "Hello to
FullCircle Magazine
Readers!!"
```

This simply prints in the terminal window the message “Hello to FullCircle Magazine Readers!!”

Finally, we instantiate the Tk class - our App class - and run the main loop.

```
root = Tk()

app = App(root)

root.mainloop()
```

Give it a try. Now things actually do something. But again, the window position is very inconvenient. Let's fix that in our next example.

Our Third Example

Save the last example as example3.py. Everything is exactly the same except for one line. It's at the bottom in our main routine calls. I'll show you those lines with our new one:

```
root = Tk()

root.geometry('150x75+550+150
')

app = App(root)

root.mainloop()
```

What this does is force our initial window to be 150 pixels wide and 75 pixels high. We also want the upper left corner of the window to be placed at X-pixel position 550 (right and left) and the Y-pixel position at 150 (top to botton). How did I come up with these numbers? I started with some reasonable values and tweaked them from there. It's a bit of a pain in the neck to do it this way, but the results are better than not doing it at all.

Our Fourth Example - A Simple Calculator

Now, let's look at something a bit more complicated. This time, we'll create a simple “4 banger” calculator. If you don't know, the phrase “4 banger” means four functions: Add, Subtract, Multiply, and Divide. Right is what it looks like in simple text form.

```
-----------------
|             0 |
-----------------
| 1 | 2 | 3 | + |
-----------------
| 4 | 5 | 6 | - |
-----------------
| 7 | 8 | 9 | * |
-----------------
| - | 0 | . | / |
-----------------
|       =       |
-----------------
|     CLEAR     |
-----------------
```

We'll dive right into it and I'll explain the code (middle right) as we go.

```
from Tkinter import *

def StartUp():
    global val, w, root
    root = Tk()
    root.title('Easy Calc')
    root.geometry('247x330+469+199')
    w = Calculator(root)
    root.mainloop()
```

Outside of the geometry statement, this (left) should be pretty easy for you to understand by now. Remember, pick some reasonable values, tweak them, and then move on.

We begin our class definition and set up our __init__ function. We set up three variables as follows:

- CurrentValue – Holds the current value that has been input into the calculator.
- HolderValue – Holds the value that existed before the user clicks a function key.

```
class Calculator():
    def __init__(self,root):
        master = Frame(root)
        self.CurrentValue = 0
        self.HolderValue = 0
        self.CurrentFunction = ''
        self.CurrentDisplay = StringVar()
        self.CurrentDisplay.set('0')
        self.DecimalNext = False
        self.DecimalCount = 0
        self.DefineWidgets(master)
        self.PlaceWidgets(master)
```

contents ^

• CurrentFunction – This is simply a “bookmark” to note what function is being dealt with.

Next, we define the CurrentDisplay variable and assign it to the StringVar object. This is a special object that is part of the Tkinter toolkit. Whatever widget you assign this to automatically updates the value within the widget. In this case, we will be using this to hold whatever we want the display label widget to... er... well... display. We have to instantiate it before we can assign it to the widget. Then we use the built in 'set' function. We then define a boolean variable called DecimalNext, and a variable DecimalCount, and then call the DefineWidgets function, which creates all the widgets, and then call the PlaceWidget function, which actually places them in the root window.

```
def
DefineWidgets(self,master):

self.lblDisplay =
Label(master,anchor=E,relief
=
SUNKEN,bg="white",height=2,te
xtvariable=self.CurrentDispla
y)
```

Now, we have already defined a label earlier. However, this time we are adding a number of other attributes. Notice that we aren't using the 'text' attribute. Here, we assign the label to the parent (master), then set the anchor (or, for our purposes, justification) for the text, when it gets written. In this case, we are telling the label to justify all text to the east or on the right side of the widget. There is a justify attribute, but that's for multiple lines of text. The anchor attribute has the following options... N, NE, E, SE, S, SW, W, NW and CENTER. The default is CENTER. You should think compass points for these. Under normal circumstances, the only really usable values are E (right), W (left), and Center.

Next, we set the relief or visual style of the label. The “legal” options here are FLAT, SUNKEN, RAISED, GROOVE, and RIDGE. The default is FLAT if you don't specify anything. Feel free to try the other combinations on your own after we're done. Next, we set the background (bg) to white in order to set it off from the rest of the window a bit. We set the height to 2 (which is two text lines high, not in pixels), and finally assign the variable we just defined a moment ago (self.CurrentDisplay) to the textvariable attribute. Whenever the value of self.CurrentDisplay changes, the label will change its text to match automatically.

```
self.btn1 = Button(master, text = '1',width = 4,height=3)
self.btn1.bind('<ButtonRelease-1>', lambda e: self.funcNumButton(1))
self.btn2 = Button(master, text = '2',width = 4,height=3)
self.btn2.bind('<ButtonRelease-1>', lambda e: self.funcNumButton(2))
self.btn3 = Button(master, text = '3',width = 4,height=3)
self.btn3.bind('<ButtonRelease-1>', lambda e: self.funcNumButton(3))
self.btn4 = Button(master, text = '4',width = 4,height=3)
self.btn4.bind('<ButtonRelease-1>', lambda e: self.funcNumButton(4))
```

Shown above, we'll create some of the buttons.

I've shown only 4 buttons here. That's because, as you can see, the code is almost exactly the same. Again, we've created buttons earlier in this tutor, but let's take a closer look at what we are doing here.

We start by defining the parent (master), the text that we want on the button, and the width and height. Notice that the width is in characters and the height is in text lines. If you were doing a graphic in the button, you would use pixels to define the height and width. This can become a bit confusing until you get your head firmly wrapped around it. Next, we are setting the bind attribute. When we did the buttons in the previous examples, we used the 'command=' attribute to define what function should be called when the user clicks the button. This time, we are using the '.bind' attribute. It's almost the same thing, but this is an easier way to do it, and to pass information to the callback routine that is static. Notice that here we are using '<ButtonRelease-1>' as the trigger for the bind. In this case, we want to make sure that it's only after the user clicks AND releases the left mouse button that we make our callback. Lastly, we define the callback we want to call, and what we are going to pass to it. Now, those of you who are astute (which is each and every one of you) will notice something new. The 'lambda e:' call.

In Python, we use Lambda to define anonymous functions that will appear to interpreter as a valid statement. This allows us to put multiple segments into a single line of code. Think of it as a mini function. In this case, we are setting up the name of the callback function and the value we want to send as well as the event tag (e:). We'll talk more about Lambda in a later article. For now, just follow the example.

I've given you the first four buttons. Copy and paste the above code for buttons 5 through 9 and button 0. They are all the same with the exception of the button name and the value we send the callback. Next steps are shown right.

The only thing that hasn't been covered before are the columnspan and sticky attributes. As I mentioned before, a widget can span more than one column or row. In this case, we are “stretching” the label widget across all four columns. That's

```
        self.btnDash = Button(master, text = '-',width = 4,height=3)
        self.btnDash.bind('<ButtonRelease-1>', lambda e: self.funcFuncButton('ABS'))
        self.btnDot = Button(master, text = '.',width = 4,height=3)
        self.btnDot.bind('<ButtonRelease-1>', lambda e: self.funcFuncButton('Dec'))
```

The btnDash sets the value to the absolute value of the value entered. 523 remains 523 and -523 becomes 523. The btnDot button enters a decimal point. These examples, and the ones below, use the callback funcFuncButton.

```
        self.btnPlus = Button(master,text = '+', width = 4, height=3)
        self.btnPlus.bind('<ButtonRelease-1>', lambda e: self.funcFuncButton('Add'))
        self.btnMinus = Button(master,text = '-', width = 4, height=3)
        self.btnMinus.bind('<ButtonRelease-1>', lambda e:
self.funcFuncButton('Subtract'))
        self.btnStar = Button(master,text = '*', width = 4, height=3)
        self.btnStar.bind('<ButtonRelease-1>', lambda e: self.funcFuncButton('Multiply'))
        self.btnDiv = Button(master,text = '/', width = 4, height=3)
        self.btnDiv.bind('<ButtonRelease-1>', lambda e: self.funcFuncButton('Divide'))
        self.btnEqual = Button(master, text = '=')
        self.btnEqual.bind('<ButtonRelease-1>', lambda e: self.funcFuncButton('Eq'))
```

Here are the four buttons that do our math functions.

```
        self.btnClear = Button(master, text = 'CLEAR')
        self.btnClear.bind('<ButtonRelease-1>', lambda e: self.funcClear())
```

Finally, here is the clear button. It, of course, clears the holder variables and the display. Now we place the widgets in the PlaceWidget routine. First, we initialize the grid, then start putting the widgets into the grid. Here's the first part of the routine.

```
    def PlaceWidgets(self,master):
        master.grid(column=0,row=0)
        self.lblDisplay.grid(column=0,row=0,columnspan = 4,sticky=EW)
        self.btn1.grid(column = 0, row = 1)
        self.btn2.grid(column = 1, row = 1)
        self.btn3.grid(column = 2, row = 1)
        self.btn4.grid(column = 0, row = 2)
        self.btn5.grid(column = 1, row = 2)
        self.btn6.grid(column = 2, row = 2)
        self.btn7.grid(column = 0, row = 3)
        self.btn8.grid(column = 1, row = 3)
        self.btn9.grid(column = 2, row = 3)
        self.btn0.grid(column = 1, row = 4)
```

contents ^

```
    self.btnDash.grid(column = 0, row = 4)
    self.btnDot.grid(column = 2, row = 4)
    self.btnPlus.grid(column = 3,row = 1)
    self.btnMinus.grid(column = 3, row = 2)
    self.btnStar.grid(column = 3, row = 3)
    self.btnDiv.grid(column=3, row = 4)
    self.btnEqual.grid(column=0,row=5,columnspan = 4,sticky=NSEW)
    self.btnClear.grid(column=0,row=6,columnspan = 4, sticky = NSEW)
```

what the "columnspan" attribute does. There's a "rowspan" attribute as well. The "sticky" attribute tells the widget where to align its edges. Think of it as how the widget fills itself within the grid. Above left is the rest of our buttons.

Before we go any further let's take a look at how things will work when the user presses buttons.

Let's say the user wants to enter 563 + 127 and get the answer. They will press or click (logically) 5, then 6, then 3, then the "+," then 1, then 2, then 7, then the "=" buttons. How do we handle this in code? We have already set the callbacks for the number buttons to the funcNumButton function. There's two ways to handle this. We can keep the information entered as a string and then when we need to convert it into a number, or we can keep it as a number the entire time. We will use the latter method. To do this, we will keep the value that is already there (0 when we start) in a variable called "self.CurrentValue", When a number comes in, we take the variable, multiply it by 10 and add the new value. So, when the user enters 5, 6 and 3, we do the following...

```
User clicks 5 – 0 * 10 + 5
(5)

User clicks 6 – 5 * 10 + 6
(56)

User clicks 3 – 56 * 10 + 3
(563)
```

Of course we then display the "self.CurrentValue" variable in the label.

Next, the user clicks the "+" key. We take the value in "self.CurrentValue" and place it into the variable "self.HolderValue," and reset the "self.CurrentValue" to 0. We then repeat the process for the clicks on 1, 2 and 7. When the user clicks the "=" key, we then add the values in "self.CurrentValue" and "self.HolderValue", display them, then clear both variables to continue.

Above is the code to start defining our callbacks.

```
    def funcNumButton(self,val):
        if self.DecimalNext == True:
            self.DecimalCount += 1
            self.CurrentValue = self.CurrentValue + (val * (10**-self.DecimalCount))
        else:
            self.CurrentValue = (self.CurrentValue * 10) + val
        self.DisplayIt()
```

The "funcNumButton routine receives the value we passed from the button press. The only thing that is different from the example above is what if the user pressed the decimal button ("."). Below, you'll see that we use a boolean variable to hold the fact they pressed the decimal button, and, on the next click, we deal with it. That's what the "if self.DecimalNext == True:" line is all about. Let's walk through it.

The user clicks 3, then 2, then the decimal, then 4, to create "32.4". We handle the 3 and 2 clicks through the "funcNumButton" routine. We check to see if self.DecimalNext is True (which in this case it isn't until the user clicks the "." button). If not, we simply multiply the held value (self.CurrentValue) by 10 and add the incoming value. When the user clicks the ".", the callback "funcFuncButton" is called with the "Dec" value. All we do is set the boolean variable

contents ^

“self.DecimalNext” to True. When the user clicks the 4, we will test the “self.DecimalNext” value and, since it's true, we play some magic. First, we increment the self.DecimalCount variable. This tells us how many decimal places we are working with. We then take the incoming value, multiply it by (10**-self.DecimalCount). Using this magic operator, we get a simple “raised to the power of” function. For example 10**2 returns 100. 10**-2 returns 0.01. Eventually, using this routine will result in a rounding issue, but for our simple calculator, it will work for most reasonable decimal numbers. I'll leave it to you to work out a better function. Think of this as your homework for this month.

The “funcClear” routine simply clears the two holding variables, then sets the display.

```
def funcClear(self):

self.CurrentValue = 0

self.HolderValue = 0

self.DisplayIt()
```

Now the functions. We've already discussed what happens with the function 'Dec'. We set this

```
    def funcFuncButton(self,function):
      if function =='Dec':
          self.DecimalNext = True
      else:
              self.DecimalNext = False
              self.DecimalCount = 0
              if function == 'ABS':
                  self.CurrentValue *= -1
                  self.DisplayIt()
```

The ABS function simply takes the current value and multiplies it by -1.

```
              elif function == 'Add':
                  self.HolderValue = self.CurrentValue
                  self.CurrentValue = 0
                  self.CurrentFunction = 'Add'
```

The Add function copies “self.CurrentValue” into “self.HolderValue”, clears “self.CurrentValue”, and sets the “self.CurrentFunction” to “Add”. The Subtract, Multiply and Divide functions do the same thing with the proper keyword being set in “self.CurrentFunction”.

```
              elif function == 'Subtract':
                  self.HolderValue = self.CurrentValue
                  self.CurrentValue = 0
                  self.CurrentFunction = 'Subtract'
              elif function == 'Multiply':
                  self.HolderValue = self.CurrentValue
                  self.CurrentValue = 0
                  self.CurrentFunction = 'Multiply'
              elif function == 'Divide':
                  self.HolderValue = self.CurrentValue
                  self.CurrentValue = 0
                  self.CurrentFunction = 'Divide'
```

The “Eq” function (Equals) is where the “magic” happens. It will be easy for you to understand the following code by now.

```
              elif function == 'Eq':
                  if self.CurrentFunction == 'Add':
                      self.CurrentValue += self.HolderValue
                   elif self.CurrentFunction == 'Subtract':
                      self.CurrentValue = self.HolderValue - self.CurrentValue
                   elif self.CurrentFunction == 'Multiply':
                      self.CurrentValue *= self.HolderValue
                   elif self.CurrentFunction == 'Divide':
                      self.CurrentValue = self.HolderValue / self.CurrentValue
                   self.DisplayIt()
                   self.CurrentValue = 0
                   self.HolderValue = 0
```

one up first with the "if" statement. We go to the "else," and if the function is anything else, we clear the "self.DecimalNext" and "self.DecimalCount" variables.

The next set of steps are shown on the previous page (right hand box).

The DisplayIt routine simply sets the value in the display label. Remember we told the label to "monitor" the variable "self.CurrentDisplay". Whenever it changes, the label automatically changes the display to match. We use the ".set" method to change the value.

```
def DisplayIt(self):

print('CurrentValue = {0} -
HolderValue =
{1}'.format(self.CurrentValue
,self.HolderValue))

self.CurrentDisplay.set(self.
CurrentValue)
```

Finally we have our startup lines.

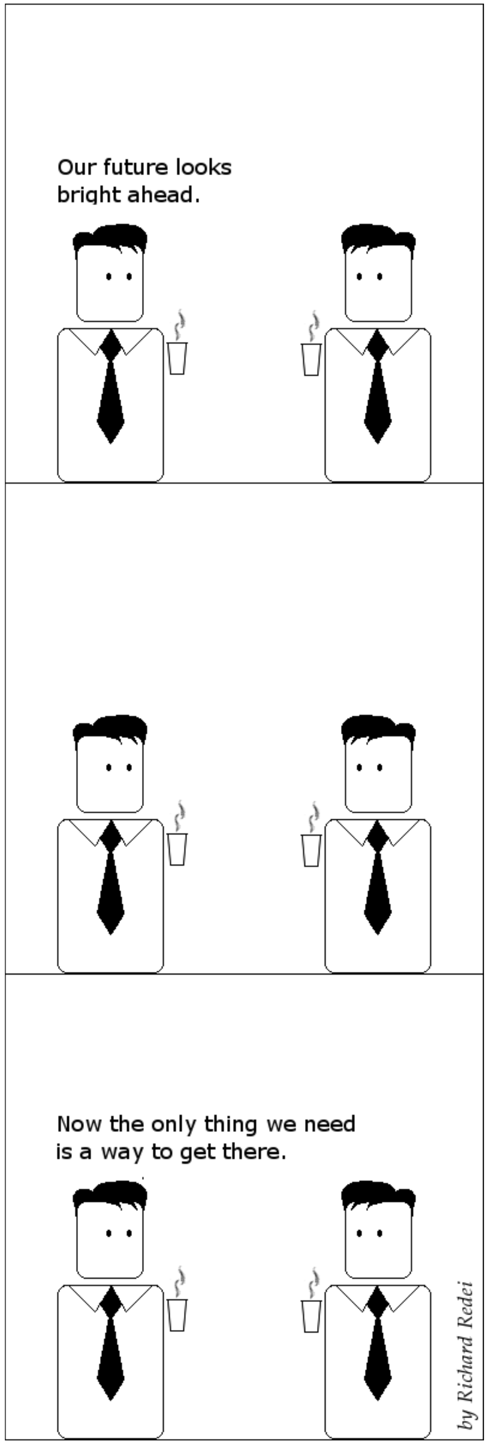
```
if __name__ == '__main__':

StartUp()
```

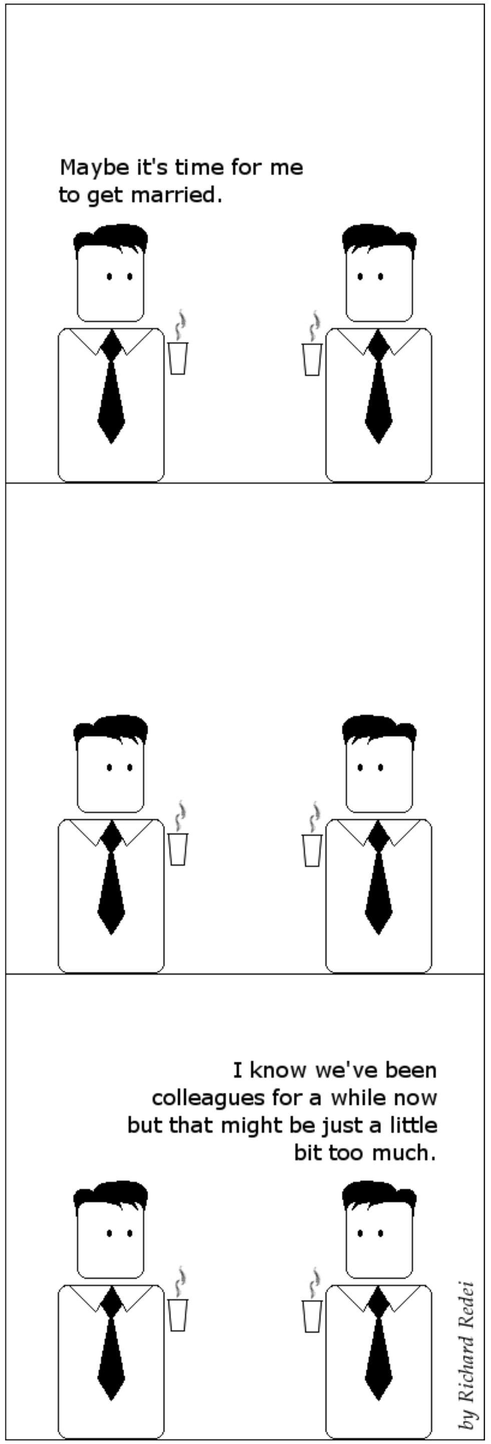
Now you can run the program and give it a test.

As always, the code for this article can be found at PasteBin. Examples 1, 2 and 3 are at: http://pastebin.com/mBAS1Umm and the Calc.py example is at: http://pastebin.com/LbMibF0u

Next month, we will continue looking at Tkinter and its wealth of widgets. In a future article, we'll look at a GUI designer for tkinter called PAGE. In the meantime, have fun playing. I think you'll enjoy Tkinter.

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


Our future looks bright ahead.
Now the only thing we need is a way to get there.
by Richard Redei

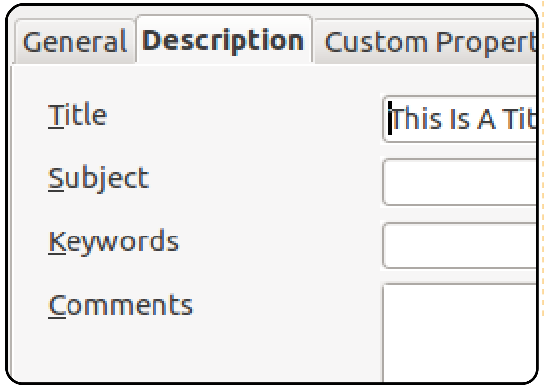

Maybe it's time for me to get married.
I know we've been colleagues for a while now but that might be just a little bit too much.
by Richard Redei

Libre Office - Part 6

Written by Elmer Perry

In this month's article, we will discover a few new ways to format our documents using page styles, headers and footers. In past articles, I have discussed the use of paragraph and character styles. Page styles are similar, but deal with the overall geometry and formatting of the entire page. Headers and footers are the area at the top and bottom of the page, and are usually the same on every page of the same style.

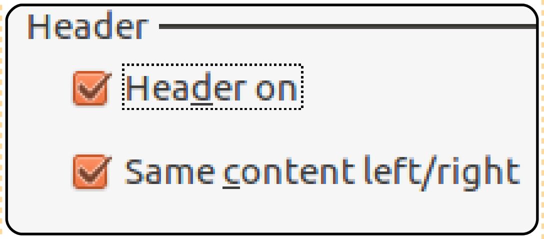
We will start by setting up our document and styles. Start a new writer document, File > New. In order to have access to the document's title, we will change some of the document's properties, File > Properties. On the description tab, put "This Is



Title	This Is A Tit
Subject	
Keywords	
Comments	

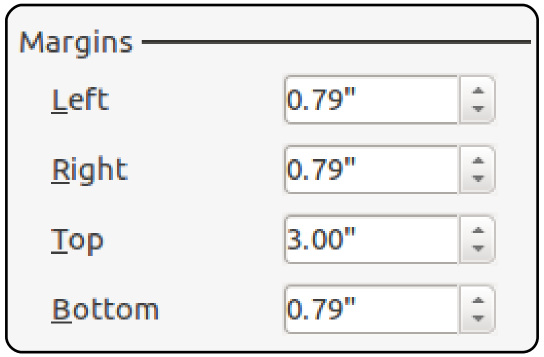
The Title" as the title of the document. We will use this later when we start creating our headers and footers. Click OK to save your changes.

Now, we need to set up our page styles. We will use three page styles, First Page, Normal Page, and Landscape. First Page and Landscape already exist, but we will modify them. We will create our Normal Page style first. For our normal page style, we want a header area at the top with a light gray background. Open the Styles and Formatting dialog, Tools > Styles and Formatting, or click on the Style and formatting button (right). Click on the page styles button (right), right-click in the window, and select new. The Page Style dialog appears. On the Organizer tab, name the style "Normal Page." Change the next style to Normal Page. This tells Writer that when we get to the end of the page, it will create a new page with the same style. On the Header tab, check Header On. This inserts a header area on the page. Still on the Header tab, click the More button. A new dialog comes up. This dialog allows us to add borders and background colors to our header. On the Background tab, pick the light gray color for the background. Click OK on both dialogs, and we are finished with our Normal Page style.



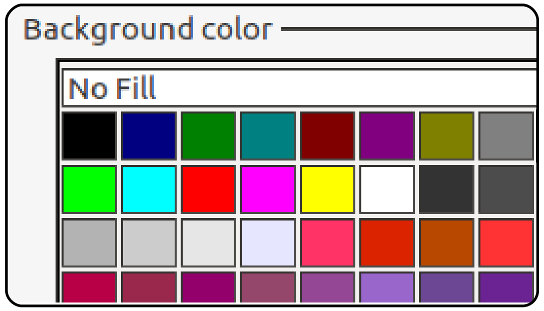
Header on ✓
Same content left/right ✓

For our First Page, we will modify the one that already exists. We want a 3" (7.5cm) margin at the top (for first page graphics added at another time), and a light gray footer area at the bottom. Right-click the First Page style in the Styles and Formatting dialog, and select modify. On the organizer tab, make our Normal Page the next style. The Page tab allows us to change the margins for the page. Make the top margin 3" (7.5cm). This time we will go to the footer tab, check Footer On, click on the More button, and choose our light gray background.



Left	0.79"
Right	0.79"
Top	3.00"
Bottom	0.79"

For our Landscape page style, we will modify the existing Landscape style. For our Landscape style, we will add both a header and footer. Right-click on the Landscape style and modify. Take a few moments to look at the page tab and notice the orientation for the page is landscape, which is exactly what we wanted. Turn on the header and footer on their respective tabs, and select the light gray background for both.



Now, we are ready to create our document. Double-click the First Page style, and the page in your document will change to the formatting we added. You will notice the light gray footer area at the bottom. Click inside the box to edit the footer. We will first add our title, Insert > Fields > Title. This inserts the title we added in the document properties. You can use this method to insert the title of the document anywhere you need it. If you change your title later in the document properties, you can update all instances of the inserted field with Tools > Update > Fields or by pressing F9 on your keyboard. Type “ Page “, remembering to put spaces on either side of the word Page, and insert the page number, Insert > Fields > Page Number. Move your cursor to the beginning of “Page” and press the tab key on your keyboard until the page number is flush against the right side of the footer area. Click out of the footer area into the main body of the page.

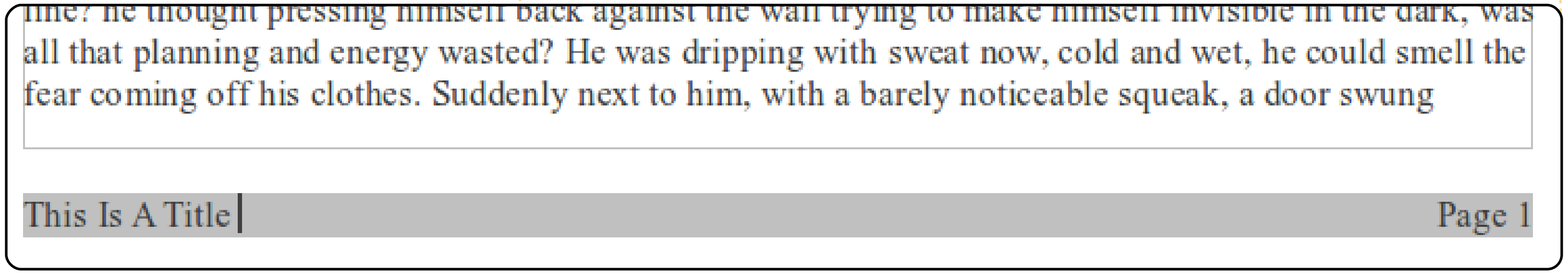
Once this is done, you can begin to type in your text. Once you reach the end of the page and a new page is inserted, you will notice it is formatted with the Normal Page style with a header area at the top. Fill in the header information just like we did for the footer of the first page. Make sure you use the fields, especially on the page number. The page number field comes in handy when we get to the third page. You will then notice the header information has been copied for you and the page number updated to reflect the current page.

Next, we will insert a Landscape page. Before you get to a new page, Insert > Manual Break. Select Page Break, and under the style, select Landscape. This will take you to a new page with a Landscape layout. Because this is a different style from our Normal Page style, we will need to fill in our header and footer information. This is handy should you need different header or footer information on some pages, just insert a page with a different page style. Once you have completed your landscaped page, create another page break (Insert > Manual Break) with a style of Normal Page. You will notice your page numbering continues, including the inserted landscaped page(s). If you do not want the inserted landscape pages included in the page count, you can manually adjust the page number in the Manual Break dialog.

Writer makes it easy to add pages with different styles and orientation, as well as automatic headers and footers.

Writer makes it easy to add pages with different styles and orientation, as well as automatic headers and footers. You can make the headers and footers as big as you want, and they can contain whatever information you want to put in them. Fields help keep certain information consistent in your document, and let you write without worrying about page numbers.

In my next article, I will move away from Writer and show you how to make a poor man's database using a Calc spreadsheet. After that, we will use our spreadsheet to create a form letter.


time? he thought pressing himself back against the wall trying to make himself invisible in the dark, was all that planning and energy wasted? He was dripping with sweat now, cold and wet, he could smell the fear coming off his clothes. Suddenly next to him, with a barely noticeable squeak, a door swung

This Is A Title | Page 1

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

Ubuntu Development Pt. 3 - Bug Fixing

Written by Daniel Holbach

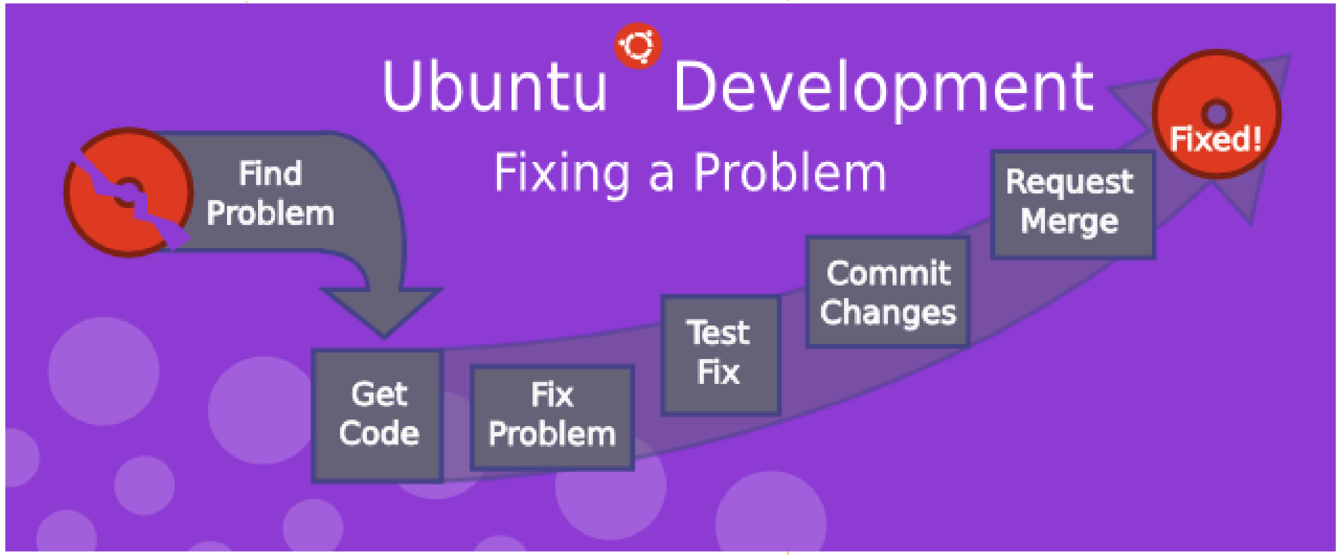
If you followed the instructions to get set up with Ubuntu Development, you should be all set and ready to go.

As you can see in the image shiwn right, there are no surprises in the process of fixing bugs in Ubuntu: you found a problem, you get the code, work on the fix, test it, push your changes to Launchpad, and ask for it to be reviewed and merged. In this guide we will go through all the necessary steps one-by-one.

Finding the problem

There are a lot of different ways to find things to work on. It might be a bug report you are encountering yourself (which gives you a good opportunity to test the fix), or a problem you noted elsewhere, maybe in a bug report.

Harvest is where we keep track of various TODO lists regarding Ubuntu development. It lists bugs that were fixed upstream or in Debian already, lists small bugs (we call them 'bitesize'), and so on. Check it out and find your first bug to work on.



Figuring out what to fix

If you don't know the source package containing the code that has the problem, but you do know the path to the affected program on your system, you can discover the source package that you'll need to work on.

Let's say you've found a bug in Tomboy, a note taking desktop application. The Tomboy application can be started by running /usr/bin/tomboy on the command line. To find the binary package containing this application, use this command:

```
apt-file find /usr/bin/tomboy
```

This would print out:

```
tomboy: /usr/bin/tomboy
```

Note that the part preceding the colon is the binary package name. It's often the case that the source package and binary package will have different names. This is most common when a single source package is used to build multiple different binary packages. To find the source package for a particular binary package, type:

```
apt-cache show tomboy | grep
^Source:
```

In this case, nothing is printed, meaning that tomboy is also the name of the binary package. An example where the source and binary package names differ is python-vigra. While that is the binary package name, the source package is actually libvigraimpex and can be found with this command (and its output):

```
apt-cache show python-vigra
| grep ^Source:
```

```
Source: libvigraimpex
```

contents ^

Getting the code

Once you know the source package to work on, you will want to get a copy of the code on your system, so that you can debug it. This is done by *branching* the source package branch corresponding to the source package. Launchpad maintains source package branches for all the packages in Ubuntu. Once you've got a local branch of the source package, you can investigate the bug, create a fix, and upload your proposed fix to Launchpad, in the form of a Bazaar branch. When you are happy with your fix, you can submit a *merge proposal*, which asks other Ubuntu developers to review and approve your change. If they agree with your changes, an Ubuntu developer will upload the new version of the package to Ubuntu so that everyone gets the benefit of your excellent fix - and you get a little bit of credit. You're now on your way to becoming an Ubuntu developer! We'll describe specifics on how to branch the code, push your fix, and request a review in the following sections.

Work on a fix

There are entire books written about finding bugs, fixing them, testing them, etc. If you are completely new to programming, try to fix easy bugs such as obvious typos first. Try to keep changes as minimal as possible and document your change and assumptions clearly.

Before working on a fix yourself, make sure to investigate if nobody else has fixed it already or is currently working on a fix. Good sources to check are:

• Upstream (and Debian) bug tracker (open and closed bugs),

• Upstream revision history (or newer release) might have fixed the problem,

• bugs or package uploads of Debian or other distributions.

If you find a patch to fix the problem, say, attached to a bug report, running this command in the source directory should apply the patch:

```
patch -p1 < ../bugfix.patch
```

Refer to the patch(1) manpage for options and arguments such as --dry-run, -p<num>, etc.

Testing the fix

To build a test package with your changes, run these commands:

```
bzr bd -- -S -us -uc
```

```
pbuilder-dist <release> build ../<package>_<version>.dsc
```

This will create a source package from the branch contents (-us -uc will just omit the step to sign the source package) and pbuilder-dist will build the package from source for whatever release you choose.

Once the build succeeds, install the package from ~/pbuilder/<release>_result/ (using sudo dpkg -i <package>_<version>.deb). Then test to see if the bug is fixed.

Documenting the fix

It is very important to document your change sufficiently so developers who look at the code in the future won't have to guess what your reasoning was and what your assumptions were. Every Debian and Ubuntu package source includes debian/changelog, where changes of each uploaded package are tracked.

The easiest way to update this is to run:

```
dch -i
```

This will add a boilerplate changelog entry for you and launch an editor where you can fill in the blanks. An example of this could be:

```
specialpackage (1.2-3ubuntu4) natty; urgency=low
  * debian/control: updated description to include frobnicator (LP: #123456)
 -- Emma Adams <emma.adams@isp.com>  Sat, 17 Jul 2010 02:53:39 +0200
```

dch should fill out the first and last line of such a changelog entry for you already. Line 1 consists of the source package name, the version number, which Ubuntu release it is uploaded to, the urgency (which almost always is 'low'). The last line always contains the name, email address and timestamp (in RFC 5322 format) of the change.

With that out of the way, let's focus on the actual changelog entry itself: it is very important to document:
• where the change was done
• what was changed
• where the discussion of the change happened

In our (very sparse) example, the last point is covered by (LP: #123456) which refers to Launchpad bug 123456. Bug reports or mailing list threads or specifications are usually good information to provide as a rationale for a change. As a bonus, if you use the LP: #<number> notation for Launchpad bugs, the bug will be automatically closed when the package is uploaded to Ubuntu.

Committing the fix

With the changelog entry written and saved, you can just run:

```
debcommit
```

and the change will be committed (locally) with your changelog entry as a commit message.

To push it to Launchpad, as the remote branch name, you need to stick to the following nomenclature:

```
lp:~<yourlpid>/ubuntu/<release>/<package>/<branchname>
```

This could, for example, be

```
lp:~emmaadams/ubuntu/natty/specialpackage/fix-for-123456
```

So, if you just run

```
bzr push
lp:~emmaadams/ubuntu/natty/specialpackage/fix-for-123456
```

```
bzr lp-open
```

you should be all set. The push command should push it to Launchpad, and the second command will open the Launchpad page of the remote branch in your browser. There, find the "(+) Propose for merging" link, and click it to get the change reviewed by somebody and included in Ubuntu.

Next month: an overview of the Debian directory structure.

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.

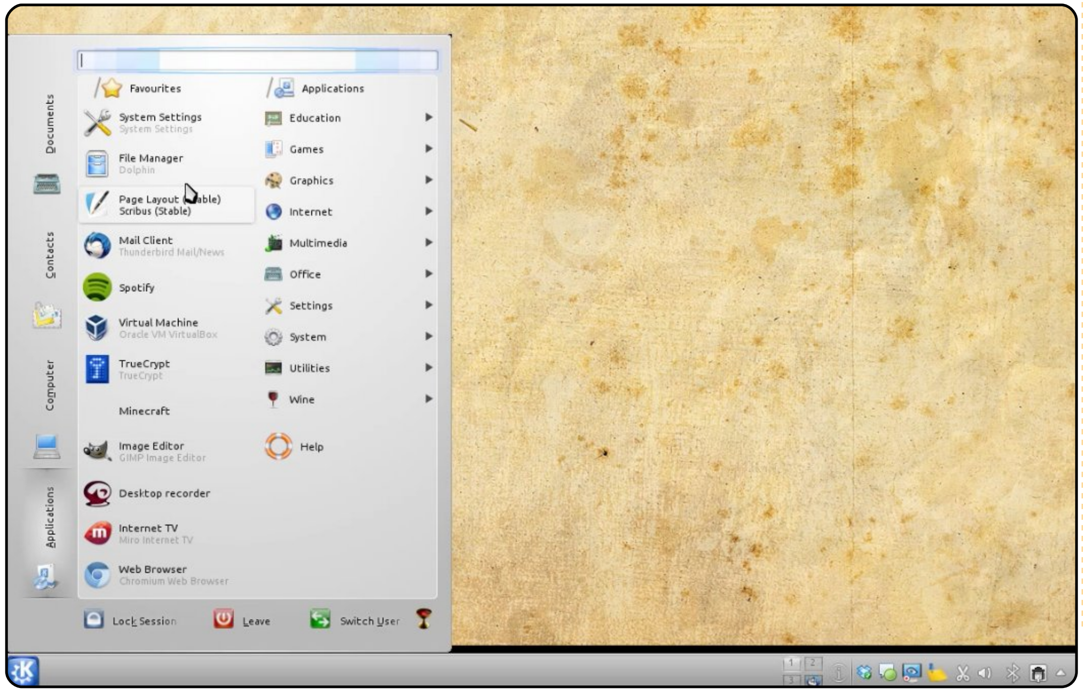
www.zerodowntime.co.uk

Written by Ronnie Tucker

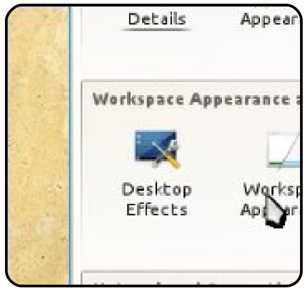
Use KDE 4.6 2 - Effects

It seems there are more KDE users out there than I thought. Quite a few people emailed me asking for a Part Two on Using KDE. So, here it is. I'll show you how to spice up your KDE desktop by enabling the KWin effects (which are to be thought of as a KDE native Compiz Fusion, but built into KDE), and the toggle effects on/off, and by editing the configuration of some effects.

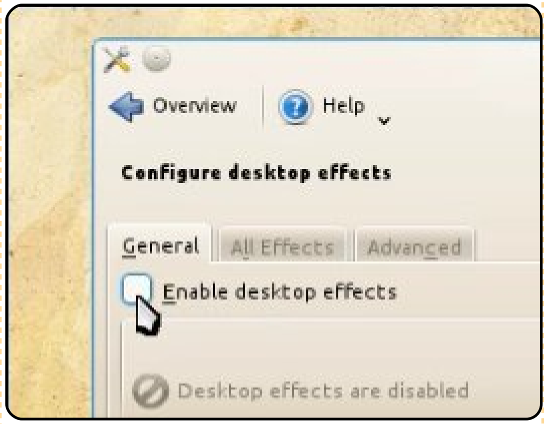
With all effects off, KDE is a bit bland:



Head into System > System Settings, and double click Desktop Effects:

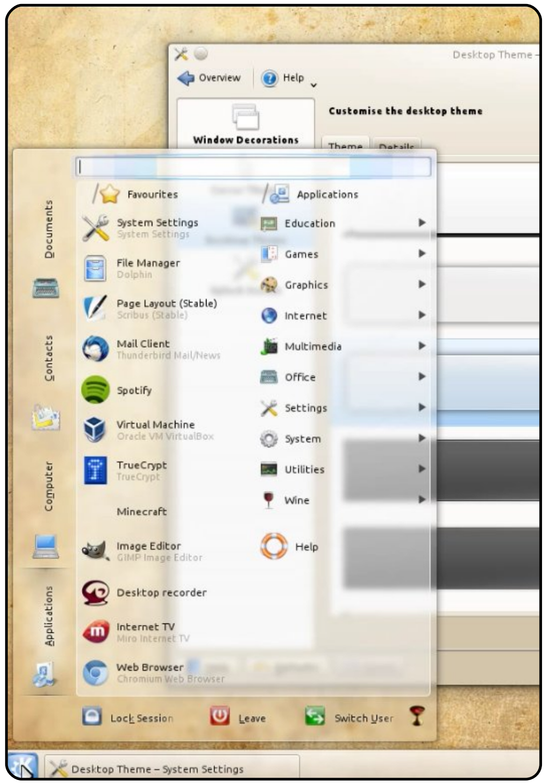


This is where the magic happens. Tick the box beside 'Enable desktop effects,' then click the 'Apply' button at the bottom right of the window (right):

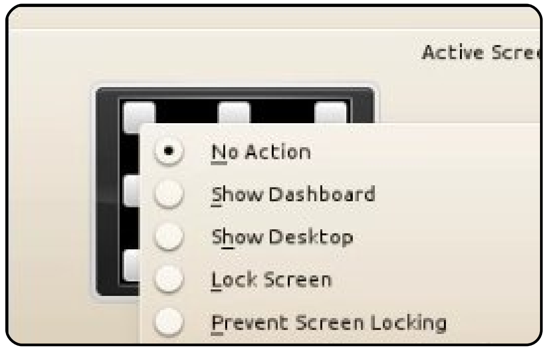


You'll get a pop-up which asks you if everything looks OK. You have several seconds to reply by clicking Accept. Your desktop effects are now active! Should your display be unable to enable desktop effects, KDE will tell you this and not black out. It's very nice that way.

Your theme probably won't show much in the way of effects, so usually I go back to the Desktop Theme window and assign/reassign a Desktop Theme. This makes sure your theme is using your snazzy new desktop effects such as blur or transparency.



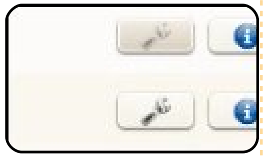
In the 'All Effects' tab (still in Desktop Effects), you'll see a list of all available effects. But the first thing I like to do is assign effects to the screen corners, which is in 'Workspace Behaviors':



I assign a desktop grid to my top left, and the cube to the top right, but you can assign them as you see fit.

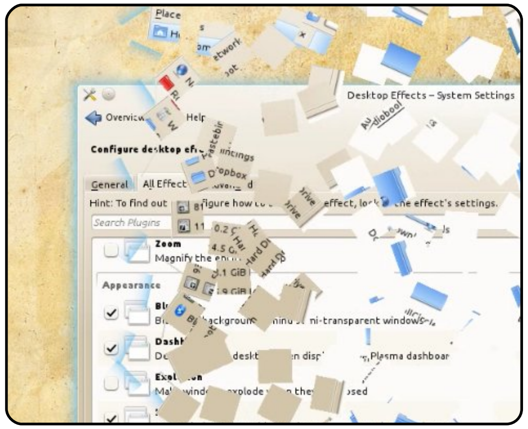
Going back into Desktop Effects (General tab), you can change how you switch windows. I, personally, prefer Flip Switch, but there are several to choose from. Below that you can change how you switch desktops. I like to slide. Below that you have the animation speeds.

Going into the 'All Effects' tab again, it's time to configure your effects. First off, the old classic Wobbly Windows. Clicking the spanner button on the right of each effect name will let you edit that effect's settings.

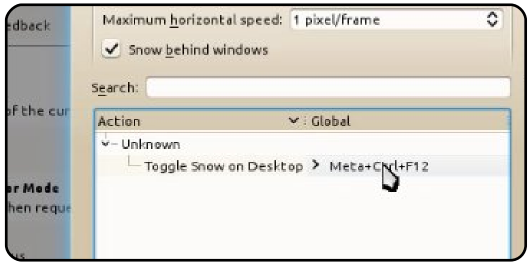


The Appearance items will let you change how a window is shown or closed. The animations range from Glide, in which the window smoothly fades in from small to large through to other animations which explode the windows into smithereens!

You can also customise the key settings to enable or disable an



effect. In this example, I assigned Ctrl+F12 to start/stop snowflakes falling on my desktop. This is done by clicking the current key binding, clicking Customise, and doing the key combination to assign it. Desktop Effects will also tell you if that key combo is being used elsewhere, and give you the option to assign that key combo to the currently selected effect, thus removing it from its previous effect.



Maximum horizontal speed: 1 pixel/frame

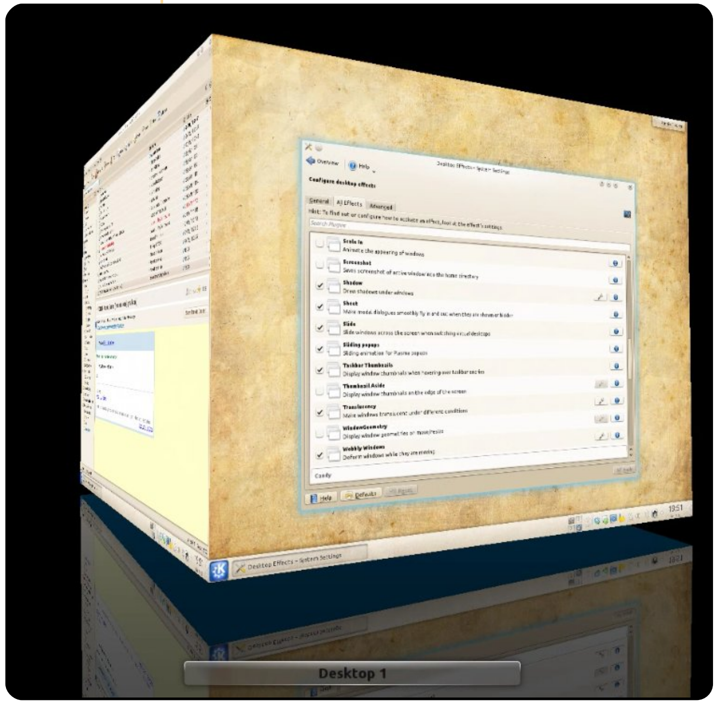
✓ Snow behind windows

Search:

Action	Global
Unknown	
Toggle Snow on Desktop	Meta+Ctrl+F12

There's a lot of things you can do with the effects that not only make your desktop look pretty, but also help you in your work with features such as dimming/blurring unselected/frozen windows, or shading the background, thus highlighting the admin login and such like.

Like last time, I recorded my desktop as I ran through this tutorial, so you can see the above effects being enabled/edited in my video on YouTube: http://www.youtube.com/watch?v=YSSE-xO9vT0

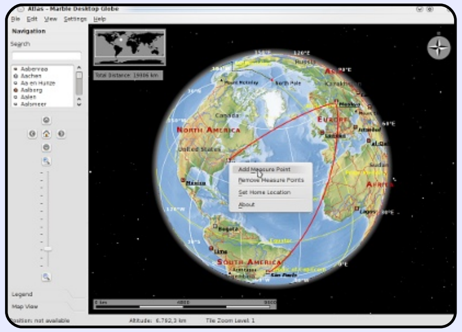


Before I leave you to play with your wobbly windows and cubes, I thought I'd give you a list of equivalent applications. It's daunting to try to find your KDE equivalent of something, so (on the next page) I've listed some of the most commonly used and installed-by-default apps in Ubuntu with their KDE kousin.

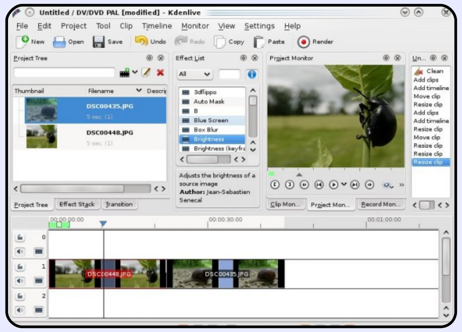
Are there any questions you have about KDE you'd like an article on? Drop me an email to ronnie@fullcirclemagazine.org and I'll see if I can make your wish come true.

Ubuntu:	**Kubuntu:**	**Purpose:**
Graphics:		
Evince	Okular	Document Viewer
gThumb	GwenView	Image Viewer
Internet:		
Evolution	Kmail	Email
Firefox	Rekonq	Browser
Pidgin	Kopete	Instant Messaging
Transmission	Ktorrent	BitTorrent
Office:		
LibreOffice	LibreOffice	Office
Sound/Video:		
Brasero	K3B	Disc Burning
Rhythmbox	Amarok	Audio
Movie Player	Dragon Player	Video
Utilities:		
Nautilus	Dolphin	File Manager
GEdit	Kate	Text Editor
Screenshot	KSnapShot	Screen Grabber
Terminal	Konsole	Command Entry
Archive Manager	Ark	File Compression

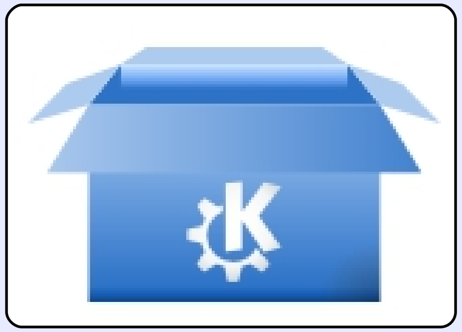
Honorable Mentions:



Marble is a Virtual Globe and World Atlas that you can use to learn more about Earth: You can pan and zoom around and you can look up places and roads. A mouse click on a place label will provide the respective Wikipedia article.



Kdenlive is an intuitive and powerful multi-track video editor, including most recent video technologies.



Kfilebox is a small application which allows quick and easy installation of the DropBox client without installing Gnome/Nautilus - http://kdropbox.deuteros.es/

Klipper is a clipboard app. The item you copied last will be the default one to be pasted, but others are stored in a buffer, so you can choose to paste your selections in a different order. It also converts URL's to barcodes.

Written by Ronnie Tucker

Write For Full Circle Magazine

Guidelines

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

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.

LINUX LAB

Written by Frank Denissen

Creating Your Own Repository

All software installed by default on a Debian-based system (like Ubuntu and Kubuntu) is organized in packages. The packages themselves are stored in a repository. The installation CD contains such a repository, but, in most cases, one accesses a repository via a server, the so-called mirror. Such a mirror gives access to a copy of the original repository created by the owner of the distribution. Any new version of a package is added to the distribution repository and afterwards copied to all mirrors.

One system (for example your PC) can obtain packages from one to many repositories. The list of repositories used by a system can be found in the files /etc/apt/sources.list and /etc/apt/sources.list.d/*.list, and can also be found under Settings in GUI packet management tools like Synaptic (Repositories) and kPackageKit (Origin of Packages).

The contents of all repositories is reread when we execute the command “apt-get update,” or when we press the Reload button in Synaptic. This allows the tools to verify which packets have new versions and offer them for upgrade.

It is also possible to create your own repository for private use.

Why would you create a private repository?

Well, I have a number of packages that are not available from the standard repositories. I downloaded packages from vendor sites containing drivers for my all-in-one scanner and my graphical card, I have some packages that are required by these driver packages and that are no longer supported by the newer Ubuntu versions and, finally, I created some packages myself.

I put any new version of such a package in my private repository. When my children come home on the weekend from university, and they push on the Reload button in Synaptic, the new package versions are nicely installed on their laptops. That is easy for me as I'm sure that any new package version will find its way to each PC without any further intervention from my side.

Procedure

The creation of a repository takes five steps:
• install the packages with the necessary tools
• create a digital signature
• create the repository directory and the related configuration files
• add packages to the repository and build the repository. Repeat this step each time you have added a new package or package version.
• make your repository known to the package tools on your system. Repeat this step for each system you manage.

In case you have multiple systems, you must decide if you want to distribute your repository to the other systems via a web-server (http) or using a directory shared via NFS or Samba.

Step 1

Install the packages ***apt-utils***, ***gzip***, ***make*** and ***gnupg***. You need additionally a web-server like ***apache2*** in case you want to make your repository accessible via the web.

Step 2

If you don't have a digital signature yet, make one now with the command:

```
gpg --gen-key
```

This tool will ask a lot of questions. The most important ones are your name, your e-mail address, and a pass-phrase. A reasonable default is proposed for the other, more difficult questions.

Step 3

Make now a directory to store the packages. This directory must finally be accessible by all your systems.

/var/www/repository is a good

choice in case you decided to use apache as a web-server. **/mnt/repository** could be used if you decided to go for NFS or Samba.

You will need the following configuration files in this directory: a public key, the steer-file for apt-ftparchive, and a makefile.

You can create the public key with:

```
gpg --export -a >
repository.gpg
```

The steer-file apt-ftparchive.conf can be created with a text editor (e.g. kate), and should have the following contents (above right, but replace "John Doe" with your own name).

Create also a makefile (below right), Makefile. Notice that all lines, except the first one, must start with a tab (not spaces !).

Step 4

Put your binary and/or source packages in this directory.

I will use as example a packet I created myself. It binds commands to the multimedia keys on my Cherry keyboard. The binary package is called ***cherry-keyboard_1.1_all.deb***. The related source packages are cherry-keyboard_1.1.dsc, cherry-keyboard_1.1_i386.changes and cherry-keyboard_1.1.tar.gz.

You may place the packages in subdirectories if you like: the apt-ftparchive tool will scan all subdirectories.

When you are ready, go to your repository directory and type "make" on the command line: your repository will be built. You will be prompted for the pass-phrase linked to your digital signature at the end of the execution.

Execute "make" again, each time you add a new package or package version. This will update the repository.

```
APT {
  FTPArchive {
    Release {
      Origin "John Doe";
      Label "John Doe";
      Suite custom;
      Codename private;
      Architecture any;
      Description "Private packages by John Doe";
    }
  }
}
```

Step 5

The last step is to make your repository known to the package tools on your PC.

Make first your public key known to apt, so that it can verify the signature of the repository files:

```
sudo cp repository.gpg
/usr/share/keyrings
```

```
sudo apt-key add
/usr/share/keyrings/repositor
y.gpg
```

```
all:
        apt-ftparchive packages . > Packages
        gzip -9 < Packages > Packages.gz
        apt-ftparchive sources . > Sources
        gzip -9 < Sources > Sources.gz
        apt-ftparchive contents . > Contents
        gzip -9 < Contents > Contents.gz
        rm  Release.gpg || true
        apt-ftparchive --config-file=apt-ftparchive.conf release . > Release
        gpg -b -o Release.gpg Release
```

contents ^

Finally, configure the location of your repository by creating a file /etc/apt/sources.list.d/repository.list, using for example sudo kate.

The contents depends on the distribution method you have chosen:

when you have exported your directory via NFS or Samba as /mnt/repository:

```
deb  file:/mnt/repository/. ./
```

```
deb-src
 file:/mnt/repository/. ./
```

when you made your repository available via web-server 192.168.0.5:

```
deb
 http://192.168.0.5/repository/. ./
```

```
deb-src
 http://192.168.0.5/repository/. ./
```

We are ready now. Verify that everything is working fine by executing the following commands:

```
sudo apt-get update
```

```
apt-cache show cherry-keyboard
```

You should now get something like:

cherry-keyboard - Enables multimedia keys on Cherry keyboard

References:
"***The Debian System - Concepts and Techniques***" by Martin F. Krafft, 2005, Open Source Press GmbH, Germany, ISBN 3-937514-07-4

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 Adel

Allow me to introduce myself. My name is Adel, and I am a Kazakh from China. (Yeah, a lot of us live in China, since China and Kazakhstan are neighboring countries. In fact, my home is just near the intersection of the four countries: Russia, China, Mongolia, and Kazakhstan.)

I met Ubuntu much later than most Full Circle readers - at version 10.04. When I rolled into college in Beijing in 2009, I didn't know much about computers, and I didn't own a computer, since it was too expensive for me. So, I often went to the netbar to enjoy the computers. The cost is 2 yuan per hour, about $0.31 US. I like collecting different software and other things while others are enjoying computer games. In fact I seldom play games.

Then I accidentally came across Ubuntu, version 10.04. At first, I just thought it was a program, but soon I found that it was so popular in the software websites. Yes, it is an operating system.

In China, nearly 100% of computers run Windows, including those in the netbars and schools. When I use a computer in the netbar, it always crashes, which often drives me mad - you never know why they do this. I wanted to buy a laptop for myself because I wanted to try Ubuntu so much, and with one I need not go to netbars, but this was impossible.

Then I found a new software called VirtualBox, which I think many of you are familiar with. With it, I could run Ubuntu in the computer in the netbar! This is exciting! Almost every weekend I went to the bar to virtualize my Ubuntu. How I wanted a computer for myself!

The things above happened in 2010, and, in December of that year, I finally bought my first computer. It has an Atom D510 CPU and 2GB DDR and a 160GB hard disk. In fact it is a netbook. The second day, I went to the netbar and downloaded the latest Ubuntu 10.10 and installed it on my computer. Yes, I installed it by clearing the whole disk which ran Windows 7 when I bought it, which is so slow when I start it.

Now I have been using Ubuntu more than 3 months. Everything is going so perfectly! I work with OpenOffice.org (can you help me install LibreOffice? thank you!), watch videos with VLC, and chat with Skype since it allows me to do video chatting. Firefox works very well too, and I also installed Chrome. Almost forget one thing - I also installed Macbuntu. I think you must have heard of it? It is so wonderful not only because it looks beautiful, but also you can use “expose” by moving the mouse to the right corner or left corner when you are dealing with many apps. (I had once used my friend's Mac OS, but the expose in it is much less convenient than Macbuntu, for you must use the keyboard and four fingers to use it, not just the mouse.)

Yeah, that's all my wonderful experience with Ubuntu. Nearly all my friends envy my desktop when they see my computer. (They just think it is Mac OS, hahahaha.) I love it so, so, so much!!! I will never leave it and never use that heavy and slow Windows!

Now my life has really, really changed. With color everyday, Ubuntu gives me so much pleasure and excitement, although I know little about the Linux world. No, Ubuntu shouldn't be treated as Linux, it is Ubuntu, the way it is, which always gives you new convenience, new choices, and new adventure.

Thanks for reading my letter. My English level is just OK since it is not my mother language. (I started learning English in 2003.) I will be very happy if you give me a short reply that says you have spared your time to read my article, no matter if you print it or not. Maybe, I should write this in Chinese and send it directly to the Chinese part, but, after thinking about it, I determined to send it to you since I want you and foreign friends to know my confident and happy experience with Ubuntu.

contents ^

Write Something. Write Anything!

Written by Allan J. Smithie

Have you ever sat there, thinking "I wish I could write?" Not "I wish I could write like Joe Schmo or anyone in particular," just "I wish I could write." I think you can. I don't mean the one novel that's in all of us. Believe me, I've read a lot of those. It's not true.

But you can write something. An opinion? Opinions are good. We all have them. Done something technical? What about a how-to, or a review? Maybe a poem? A haiku in response to Eric Schmidt? Go ahead, learn a new art-form.

I write a lot of words, and I confess I have followed the rules and I have failed. Also, I have thrown away the rules and I have failed. That's why we have editors.

I can give you lists of attributes of good writing:

- Clarity
- Accuracy
- Relevance
- Sincerity (Remember, if you can fake this, you can fake anything.)
- Concision
- Transparency
- Consistency

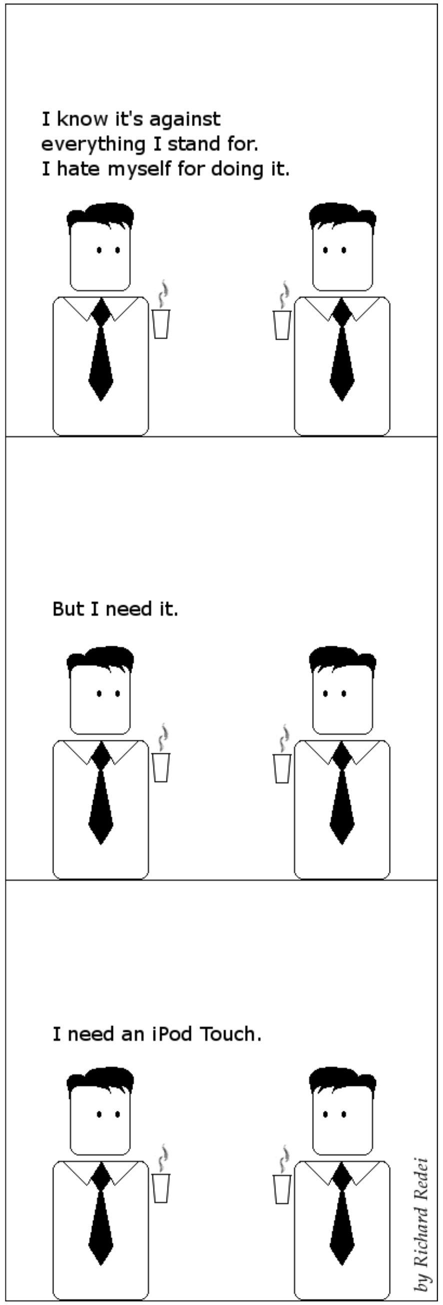
I can give you list upon list of rules for writing most kinds of text. You know the kind of thing I mean:

- Put the reader first.
- Be clear.
- Be specific.
- Get to the point. Then stop.
- Express one thought at a time.
- Use short phrases.
- Use short sentences.
- Use short paragraphs.
- Never use a long word when a short one will do.
- Edit thoroughly; cut, cut, cut

I know it's difficult. Web readers have a short attention span, and you just can't show off the extent of your vocabulary, given our readership is trans-national. That doesn't make it impossible.

I mark my own papers these days: 'must try harder'.

Why don't you give it a go? Any subject will do. Go ahead, write something. Write anything.

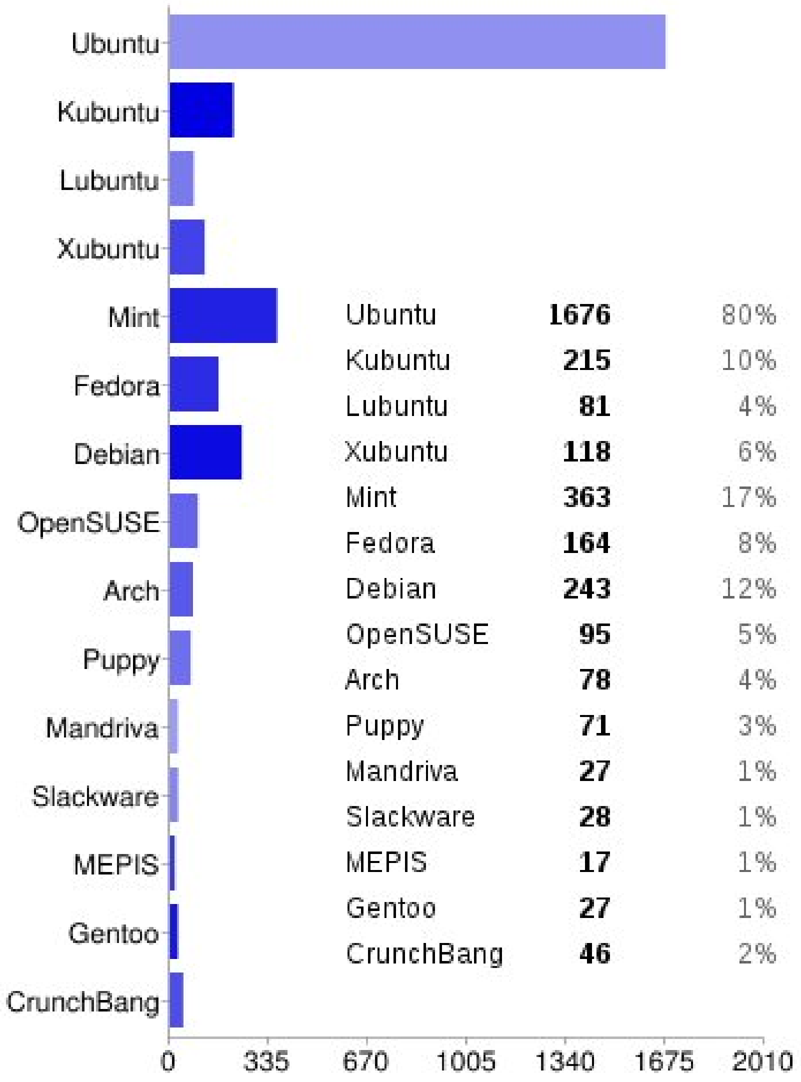

I know it's against everything I stand for. I hate myself for doing it.
But I need it.
I need an iPod Touch.
by Richard Redei

I THINK...

Last month's question was:

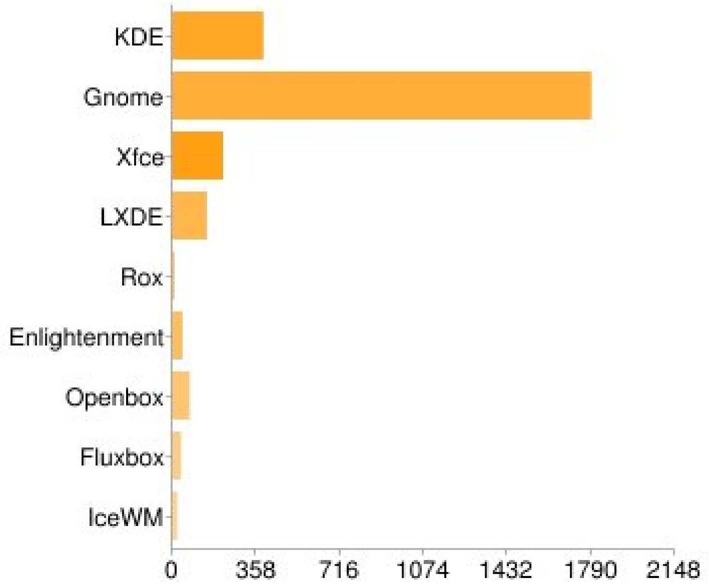
What distro(s) do you use?

What distro(s) do you use?



Ubuntu	**1676**	80%
Kubuntu	**215**	10%
Lubuntu	**81**	4%
Xubuntu	**118**	6%
Mint	**363**	17%
Fedora	**164**	8%
Debian	**243**	12%
OpenSUSE	**95**	5%
Arch	**78**	4%
Puppy	**71**	3%
Mandriva	**27**	1%
Slackware	**28**	1%
MEPIS	**17**	1%
Gentoo	**27**	1%
CrunchBang	**46**	2%

... and with those distro(s), which desktop environment(s) do you use?



KDE	**388**	19%
Gnome	**1791**	87%
Xfce	**217**	11%
LXDE	**146**	7%
Rox	**5**	0%
Enlightenment	**43**	2%
Openbox	**72**	4%
Fluxbox	**36**	2%
IceWM	**21**	1%

contents ^

Ronnie says: Even though I purposely didn’t mention Unity, I still ended up with a ton of anti-Unity comments. So, Ignoring the negative comments towards Unity, you thought:

LXDE is one of the best desktop environments out there. I even use it on my gaming rig with 8 GB of RAM and a hexacore processor. KDE is my choice when I want eye-candy and productivity.

Bodhi for the win!

Kubuntu on my desktop & laptop for work on an everyday basis. Gnome for some short time basis usage, and Lubuntu/CrunchBang for low resources machines.

KDE made huge steps but it seems too bloated.

I decide what I use, and I decide what apps I need. I need only a tiling wm and terminals + firefox + geany. Nothing more and nothing less.

Mint KDE gives me the joy of Linux, the beauty and ease of use of KDE, plus a true ready-to-use out-of-the-box experience. The fact that they don't release updates before they're READY? Priceless.

Linux Mint works 'Out of the Box'. Nothing to add or download; just turn it on and use it.

Xfde and Enlightenment FTW!

Ubuntu rescued my computer that wouldn't run Windows XP any more. Love it!

Ubuntu 11.04 Unity 2D in a laptop, and Ubuntu 8.04 in a Dell MIni 9, and Vista in a PC.

LXDE rocks! Very low RAM use.

OpenSUSE & KDE, Rules!!!

Most of the time I use Ubuntu, but in slow machines I use Xubuntu or Lubuntu.

I have multiple machines, 1 Fedora running KDE, and 2 Ubuntu running Gnome. I also have a couple headless servers but I didn't count those.

Waiting for KDE4- LTS distro supporting my Wacom Bamboo CTH-460 pen tablet.

Ubuntu 11.04/Unity for my main home PC. Ubuntu 10.04, no desktop, for media center. Headless Ubuntu 11.04 server for all home server needs. Lubuntu 11.04 for my old netboox Asus eeepc 900.

I really hope KDE will be the next hit. It just keeps getting better and better.

Ubuntu and Gnome to show off, Mint and LXDE for work, and Puppy for troubleshooting.

Lubuntu on my old laptop works great. I prefer it over Xubuntu. At office, Ubuntu 10.04 LTS.

I will use the Classic Desktop of Ubuntu 11.04 until summer 2012. Till that time the GNOME-Shell and Unit should be cured of childhood diseases. Then I will decide to go forward with the GNOME-Shell, Unity ... or even KDE?

I am using Unity right now on my Ubuntu machine. But I am thinking about give another chance to Gnome 3 Shell... However, as both of them are sometimes acting funny on Ubuntu, I have also a Fluxbox session, just in case... My other (older) computer is happily running Xubuntu.

The question I'd like to pose for FCM#52 is:

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

To give your answer, go to: http://goo.gl/MOHnG

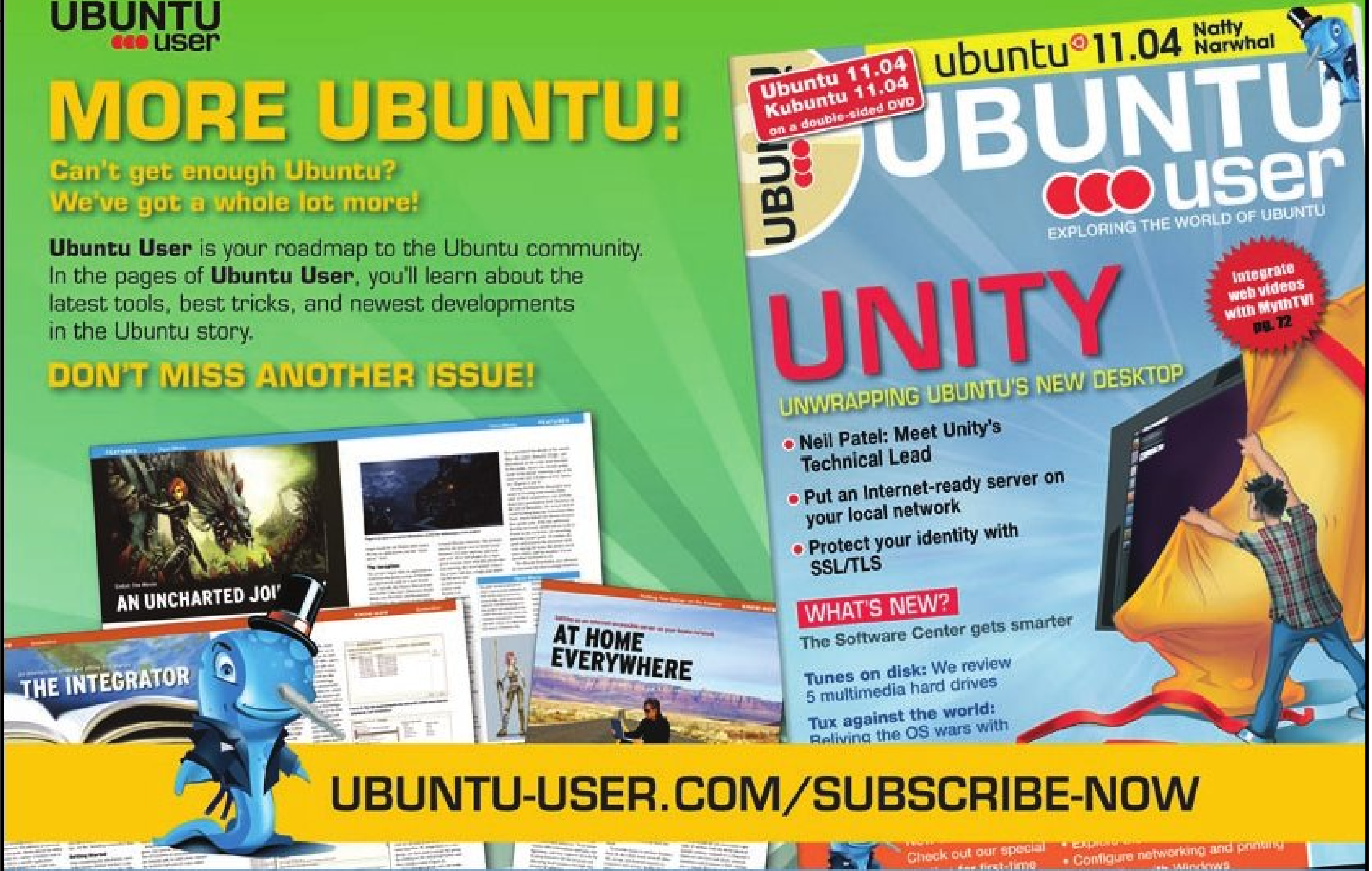
UBUNTU user

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 11.04 Natty Narwhal
Ubuntu 11.04 Kubuntu 11.04 on a double-sided DVD
UBUNTU user
EXPLORING THE WORLD OF UBUNTU
UNITY
UNWRAPPING UBUNTU'S NEW DESKTOP
Integrate web videos with MythTV! pg. 72
Neil Patel: Meet Unity's Technical Lead
Put an Internet-ready server on your local network
Protect your identity with SSL/TLS
WHAT'S NEW?
The Software Center gets smarter
Tunes on disk: We review 5 multimedia hard drives
Tux against the world: Reliving the OS wars with
AN UNCHARTED JOI
THE INTEGRATOR
AT HOME EVERYWHERE
Check out our special
Configure networking and printing

UBUNTU-USER.COM/SUBSCRIBE-NOW

FOLLOW US ON TWITTER: UBUNTUUSER FACEBOOK: UBUNTUUSERMAG

Written by Dave Rowell

GRAMPS

If you're at all interested in family history or genealogy, you'll need to use some computer program to keep track of the information you'll accumulate. Every time you try to change or update something, you'll find that more than one 'document' will be involved – you'll have to find and update them all! A program will facilitate updating and make sure the update is propagated wherever it's needed.

If you're running Linux as your OS, there's not a lot of choice – install Gramps! Fortunately, Gramps is mature, stable, easy to use, and very capable. Gramps is actively maintained by a dedicated and very responsive group of developers. Let's see how it stacks up to Family Chronicle's old list of required features:

• **Data Integrity** – Gramps doesn't change or add to the information you supply, as some other programs do. In another meaning of integrity, Gramps uses good database technology to keep your data safe.

• **Name and Date Recording** – Gramps has more than adequate provision for entering names and dates, thanks to an international group of developers and users. Dates are entered in your choice of format and calendar. But remember, the usual date format in genealogical circles is day, month, year.

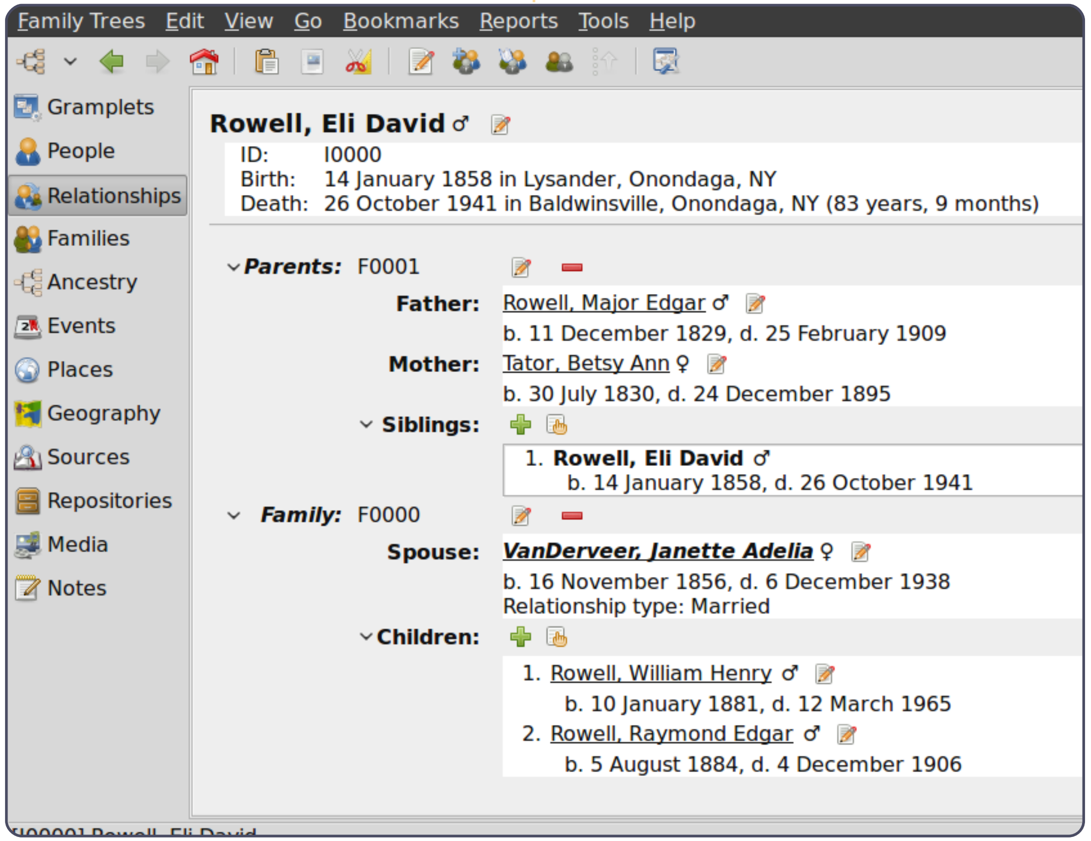
• **Place Recording** - Gramps has an interesting set of options for entering places. All the necessary fields such as Street, City, County, State, and Country are provided. The database may be sorted on any of them. Additionally, there is provision for entering geographical coordinates. If that is done, Gramps can display the data on a map.

• **Source Documentation** – Ya gotta do it! Gramps provides the tools to document the source for each datum you enter. And for each source, the repository where you found it. I can't stress enough how important it is to source each fact that you enter – as you enter it. In addition, there is some provision for evaluating that source. For my taste, there are so many places to enter sourcing information as to make it confusing - my main complaint.

• **To-Do Lists** – On several screens, Gramps provides for a key indicating whether further work is needed. There is a note type for a more detailed description of work to be done. You'll note that one of the main tabs is "Gramplets". Open it, and you'll gain access to a bunch of user-developed research aids. TODO was installed by default. Use it – it beats a bunch of sticky notes!

• **Event Recording** – Gramps, as I see it, is an Event-driven program to match our event-driven lives. Enter an event, such as birth, in a person's life, and there is more than adequate provision to document and view it.

• **Parent Recording** – A child may



Rowell, Eli David ♂

ID: I0000
Birth: 14 January 1858 in Lysander, Onondaga, NY
Death: 26 October 1941 in Baldwinsville, Onondaga, NY (83 years, 9 months)

Parents: F0001

Father: Rowell, Major Edgar ♂
b. 11 December 1829, d. 25 February 1909

Mother: Tator, Betsy Ann ♀
b. 30 July 1830, d. 24 December 1895

Siblings:
1. **Rowell, Eli David** ♂
b. 14 January 1858, d. 26 October 1941

Family: F0000

Spouse: ***VanDerveer, Janette Adelia*** ♀
b. 16 November 1856, d. 6 December 1938
Relationship type: Married

Children:
1. Rowell, William Henry ♂
b. 10 January 1881, d. 12 March 1965
2. Rowell, Raymond Edgar ♂
b. 5 August 1884, d. 4 December 1906

[I0000] Rowell, Eli David

be linked to multiple families such as birth family and adopted family. It can handle a case where one parent is a birth parent while the other isn't. Gramps also provides for several types of parent relationship - from married to none.

• **Multimedia may be linked to a person, event, source** - you name it. Do think about organization of your media and whether it will stand the test of time! There is something to be said for a single directory containing items, perhaps severely cropped, to be linked.

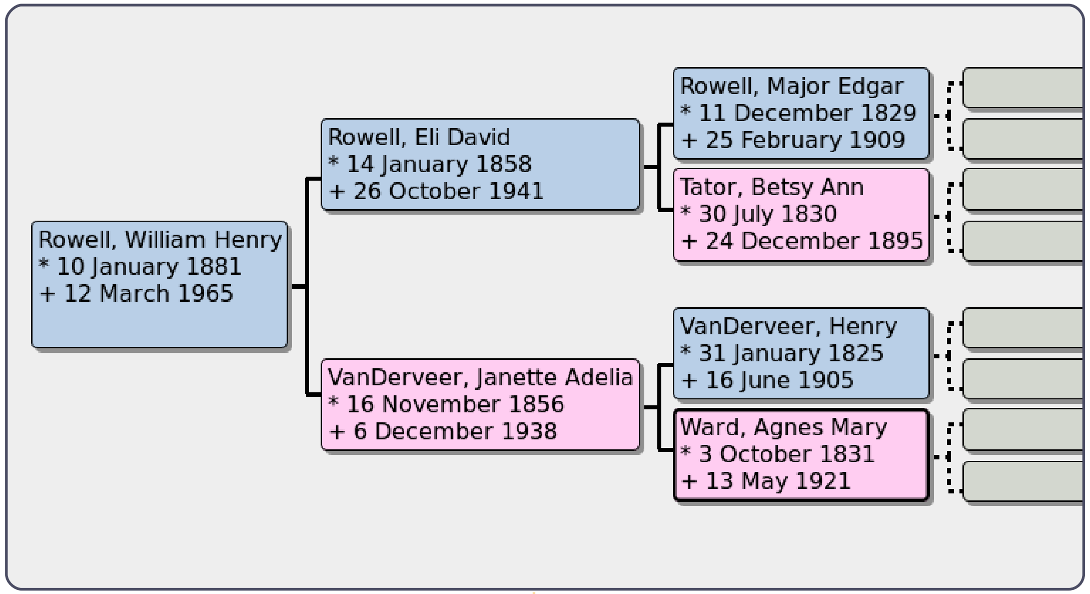
• **Data Sorting and Reporting** – Not a problem for Gramps. If sorting isn't enough for you, Gramps allows you to filter the data while you're sorting. There is provision for text reports, graphical reports, and website generation. You can generate the usual ancestor charts. I've made 18" x 24" reports for a family reunion.

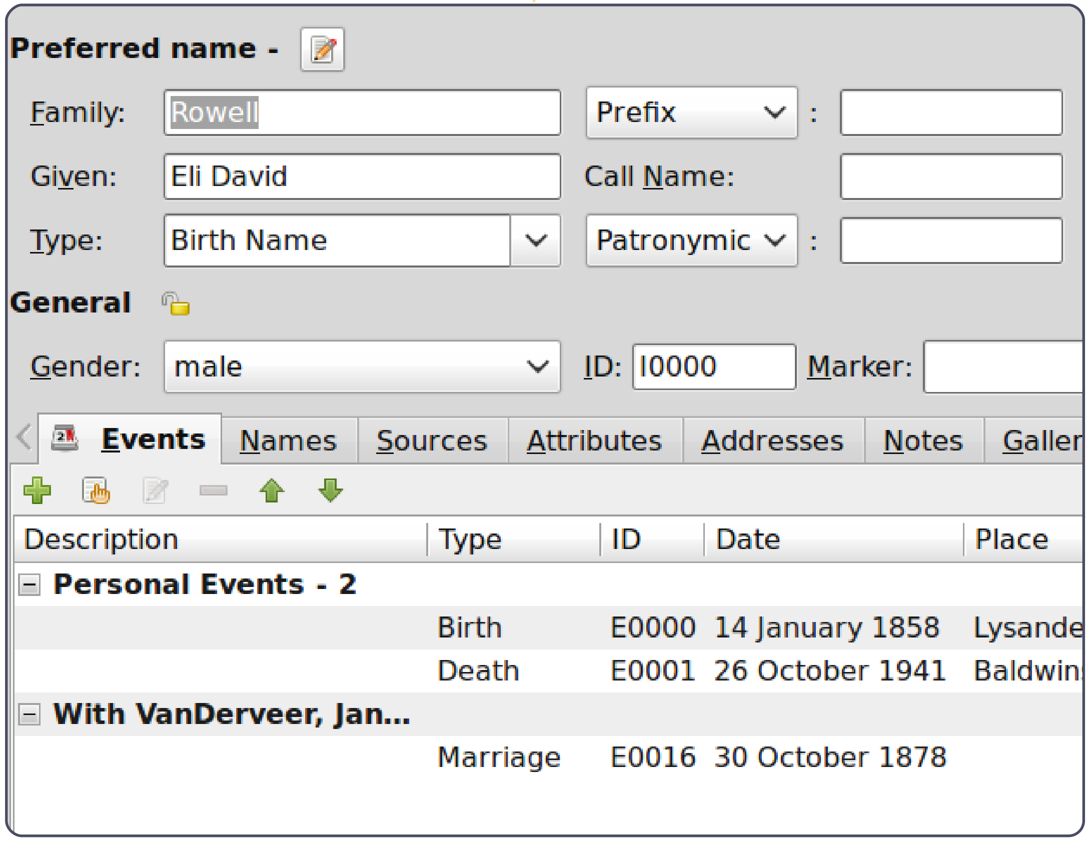
• **Back-up and Data Transfer** – Gramps has you covered here too. You can import and export GEDCOM and GeneWeb format files to transfer to and from other programs. This works as well as can be expected, but these file formats don't accommodate the bells and whistles found in other programs. Gramps provides for back-up both with and without multimedia. It works too! Interestingly, you can generate vCalendar and vCard files from your data, a feature I haven't explored.

So, as you can see, you're not crippled in any way by using Gramps. We're fortunate to have access to such a fine program.

Gramps is in the repositories and can be installed using Synaptic or, easier yet, from the Software Manager. In Linux Mint, use Menu > Software Manager (Ubuntu calls it Software Center) - type 'gramps' into the search window – click on 'Gramps' then 'Install.' Its that easy!

Next month I'll show how to get started with Gramps by starting a new database, entering your own information, and how to show your sources.




Preferred name -

Family: Rowell Prefix :
Given: Eli David Call Name:
Type: Birth Name Patronymic :

General

Gender: male ID: I0000 Marker:

Description	Type	ID	Date	Place
Personal Events - 2				
	Birth	E0000	14 January 1858	Lysande
	Death	E0001	26 October 1941	Baldwin
With VanDerveer, Jan...				
	Marriage	E0016	30 October 1878	

Every month we like to publish some of the emails we receive. If you would like to submit a letter for publication, compliment or complaint, please email it to: letters@fullcirclemagazine.org. PLEASE NOTE: some letters may be edited for space.

Kindle and Google Earth

Several months ago, I decided that I would like to have some eBooks from Amazon, so I downloaded the Kindle for Windows software, but it wouldn't install with the default version of Wine. After some searching, I found the way to get it to work on my Ubuntu 10.10 was to download the Wine 3 beta version. So far, I have had no problems with this version.

```
sudo add-apt-repository
ppa:ubuntu-wine/ppa && sudo
apt-get update && sudo apt-
get install wine1.3
```

I already had an Amazon account, so registering was not a problem, and when I updated to 11.04, I was able to get back all the books I had bought by hitting the archived button.

From 10.10 I had a problem with installing Google Earth. Downloading the .deb file from the Google site and using Gdebi to install, left me with some oversized text boxes on the screen. This was fixed by downloading the Microsoft True Type fonts:

```
sudo apt-get install
msttcorefonts
```

It seems that Goggle Earth needs them for its display.

Brian Cockley

KDE Login

I'm going to switch from Unity to KDE after trying out Unity for a while. Just dont like it. But, on another note, I wonder if you have any advice on a KDE question. I can use a Kubuntu live CD (11.04) on my desktop computer and it runs perfectly. Everything boots and I'm able to use the system. But, if I install it to the hard drive, I never get past the screen that shows those 5 icons as the desktop loads up. My system will lock up and need to be restarted.

Chris

Ronnie replies: *Having emailed back and forth with Chris about this, it seems that you have to explicitly choose KDE from the dropdown menu at login. Otherwise you'll be greeted with a blank screen.*

Pint and a Pizza

I have to admit, I am getting a little tired of the complaints about Unity, and really don't understand why they don't click a few buttons and run their Ubuntu under Gnome - it's still there you know. I am prepared to give Unity a chance, and already I am using it without thinking. In addition to that, a lot of the kinks will be ironed out by 11.10. If I didn't like Unity, I would be using Gnome and wouldn't be complaining. If I didn't like Ubuntu, I would move quietly to Kubuntu or even another distro.

And don't worry about Canonical, the owner (Mark Shuttleworth) wouldn't even notice the money missing if it folded over night. He sold his four year old company for millions when he was twenty-five and I am sure investments have doubled that by now.

So, come on people, less of the whining, and either knuckle down, or move on. Start worring about the important things in life such as Greece, our national debt, and how this will affect the price of a pint and a pizza over the coming year.

Ampers

Join us on:

facebook.com/fullcircle magazine

twitter.com/#!/fullcirclemag

linkedin.com/company/full-circle-magazine

ubuntuforums.org/forum display.php?f=270

Adding KDE

If/when you get around to writing a Part 2 article about KDE, I implore you, please make a mention on how you can switch from Ubuntu to Kubuntu without losing all your programs - that would make me a very happy bunny indeed!

John Haywood

Ronnie says: *the easiest way is to install the **kubuntu-desktop** package. You'll then be able to choose KDE, or Gnome, at login. The only down side of this is that your application menu in both KDE and Gnome will contain both Gnome and KDE applications. No big deal, it just makes your menu a bit full looking.*

More PAM

The official PAM website http://pam-face-authentication.org has the proper installation and configure procedure. The right way to install PAM in Ubuntu is described here: http://pam-face-authentication.org/downloads.php

The PPA has the .debs for Lucid, Maverick and Natty. After installing you only need to configure the plugin.

So, it's not hard to install, and the documentation is not out-of-date.

Antonio Chiurazzi

He's Right You Know

In my opinion, he [FCM#50 - My Opinion] is right. I draw parallels between Microsoft and Canonical. Microsoft changes their OS regularly to get more money from people. I assume Canonical has some reason to change things, other than just for the sake of change.

I'm a reasonably advanced user - not interested in the latest tool or trying things out for the fun of it, but just wanting a stable and consistent platform to use. I try to interest my friends in Ubuntu, especially those who don't want to shell our for the latest Windows, or MS Office, or whatever.

But more important than money is time, and most of them would rather pay for stability than to exist on the bleeding-edge where they constantly have to call me for help and advice. And frankly my time is also important - I'm not a paid support engineer for Canonical. I have a hard time recommending Ubuntu, because I know Canonical is going to fiddle just for the fun of it, and I'll be stuck with friends wanting their machine repaired.

I want a single consistent experience from release to release. I don't expect my buttons to wander around, change color, go away or anything else that hinders me from using the computer. I can deal with change, but rather that I don't want to HAVE to reset choices I have already made because some designer at Canonical thinks he knows better than I do. That's Microsoft's way of treating users.

If Canonical wants to continue to change the standards, then there should be a single configuration file of choices the user has made which will be read and obeyed by the upgrade process, containing the USER's choice of window manager, screen layout, favorite browser, commonly used tools, and so on.

My question to Canonical is, 'Is Ubuntu for the hacker, the designer, the advanced user, the "elite", the uber-geek, or is it for the masses?' If the former, then Microsoft's eternal dominance is assured. If the later, quit changing things for the sake of changing things!

Thomas

Under The Weather

Looking for some sunshine I logged onto the Met Office website and saw their desktop widget available for download. It needs Adobe Air 2.5, but the good news is that they cater for Linux users. But, this latest widget needs 1GB of RAM to run! I just upgraded my RAM to 2GB, and to think that half of it would disappear instantaneously sends shudders through my spine. I'll just keep my bookmark to the forecast page.

Roy Read

contents ^

Written by Elizabeth Krumbach

Cheri Francis Interview

This month, we're interviewing Ubuntu contributor Cheri Francis about her experience at the Ubuntu Developer Summit (UDS) in Budapest back in May.

Elizabeth Krumbach: When did you get involved with the Ubuntu community, and what areas are you currently involved in?

Cheri Francis: I got involved a few years ago, had a few forays into Ubuntu use, and then started hanging out on IRC, pitching in where I could. I am currently involved in the Ubuntu Women project, the Accessibility team, and the NGO team. I am also a member of the Ohio LoCo team, and a new member of the Ohio LoCo Council.

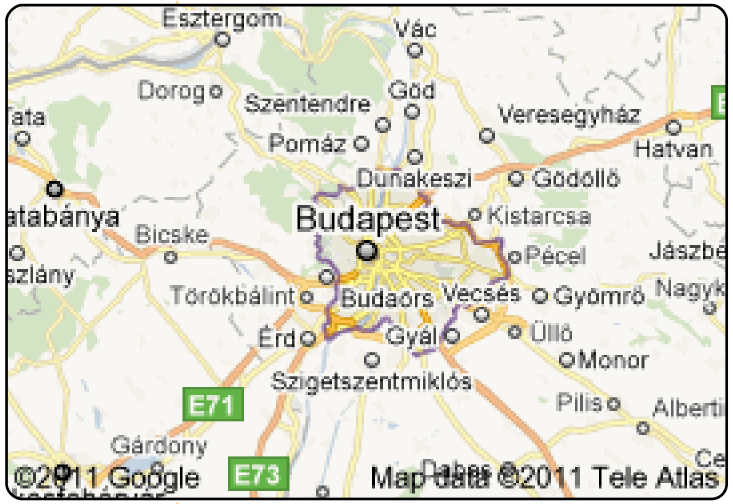
EK: What inspired you to apply for sponsorship to UDS?

CF: I got an email from a friend on the UW project whom I'd worked with in the past. I asked a few others if they thought I should, and the response was overwhelmingly positive, so I went for it. I wanted to attend because I felt it would be a great opportunity to meet some of the people I'd been working with online, as well as learning more about the nuts and bolts of how everything fits together, and the processes behind everything.

EK: What are some of the sessions you attended?

CF: I tried to focus on community and accessibility where I could. Other than that, I tried to learn as much as possible about some of the topics that were a bit over my head, but still fascinating. Attending the session about the NGO team was interesting, as I'd had no idea that it existed previously, and I think it has the potential to do amazing work. The sessions about the IRC council and Ubuntu Weekly News were very eye-opening for me. Prior to one session, I didn't know about the LoCo ISO Testing initiative, nor the Laptop Testing project. I intend to learn more about both of those.

EK: What were your biggest take-aways from the summit?



CF: It was an awesome thing, in the literal sense. So many people that give their time and energy to creating and maintaining Ubuntu, both software and community. I left UDS with the desire to do what I can to help encourage new users and contributors. By helping our LoCo be more visible and inviting, and helping to raise the visibility of my teams.

EK: Do you have any tips for first-time attendees?

CF: Ask a TON of questions before you go. Read every email multiple times to make sure you don't miss instructions. Know that the schedule is constantly changing. Don't be afraid to speak up, but try to keep it on topic.

Written by Ed Hewitt

Shadowgrounds

This month, I am continuing my series of reviews on the games included in The Humble Frozenbyte Bundle. Due to the number of games I have queued up to review in upcoming issues of Full Circle, I am bundling two games into this review.

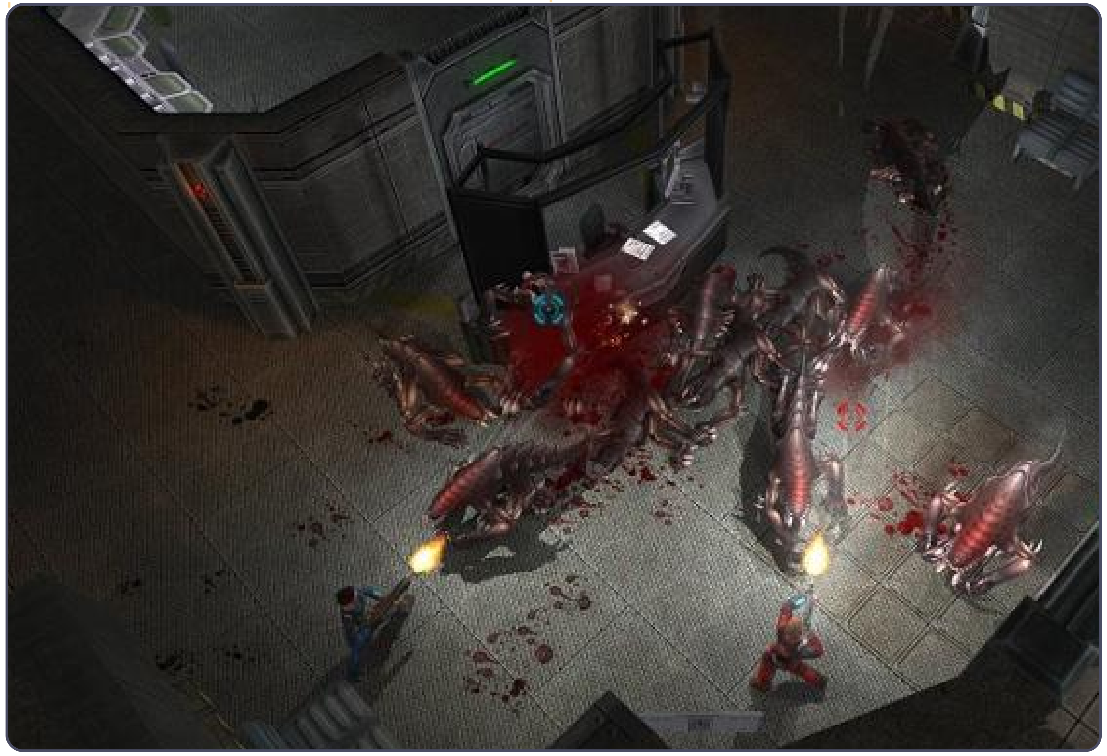
Shadowgrounds, and Shadowgrounds Survivor, are both Sci-Fi top-down shooters set on the planet of Ganymede. Both titles focus around an alien invasion of the planet and an attack on the human base. In Shadowgrounds, you play as a engineer, Wesley Tyler, who thought the game has to repair the base, fight off the alien, and escape the planet. In Shadowgrounds Survivor, the story focuses around three playable characters as they try to escape the planet. The story is solid on both titles, which is told through cut scenes and in-game video messages. Shadowgrounds features story items such as Email messages, Documents, and diary entries. All these can be picked up throughout the game, and help to expand the story further.

Shadowgrounds is a shooter, but it is unique by not using the traditional first person view. You play the whole game from a top down prospective, which works perfectly well. The keyboard is used to move the character, while the mouse is used to point your gun. It does not take long to get used to the new view, it may even encourage non-fps fans to try this shooter out.

The missions are focused on completing a series of objectives, and you will be moving around dark buildings, shooting aliens, activating objects and recovering items. Missions are enjoyable, but do become repetitive as most of the objectives are similar. What makes this game enjoyable is shooting aliens and the excellent atmosphere. Most of the game is played in the dark, with the odd flickering light. The torch attached to your gun can display light only in front of you, which adds an excellent tactical and gameplay element. You will have to be careful how you position your torch, since you could have aliens sneaking up behind you. This really does add to the tension and atmosphere. Shadowgrounds features a varied arsenal of weapons and explosives, which are very satisfying to use.

Both the graphics and sound are good in both titles, which really helps to improve the gameplay. While the graphics are dated in Shadowgrounds, with some solid improvements in Shadowgrounds Survivor, the graphics are still good by today's standards. Its standout graphic effect is the lighting from the torch. A solid soundtrack can be heard throughout, with sound effects which add to the tension and atmosphere of Shadowgrounds.

Both titles feature a decent length story campaign, Shadowgrounds boasts co-



operative play while Shadowgrounds Survivor has a Survival Mode.. Based on the style of game in Shadowgrounds, it is really begging for a Co-Op mode. While Shadowgrounds features this mode, and allows up to 4 players to play though all the missions from the story campaign, the mode has not been thought through at all. Co-Op can be played on only one computer, and not over LAN or Internet. For most people, this makes Co-Op completely pointless. Shadowgrounds Survivor has a far better extra mode, in the form of Survival Mode, which sends hordes of aliens your way, and you have to survive for as long as possible.

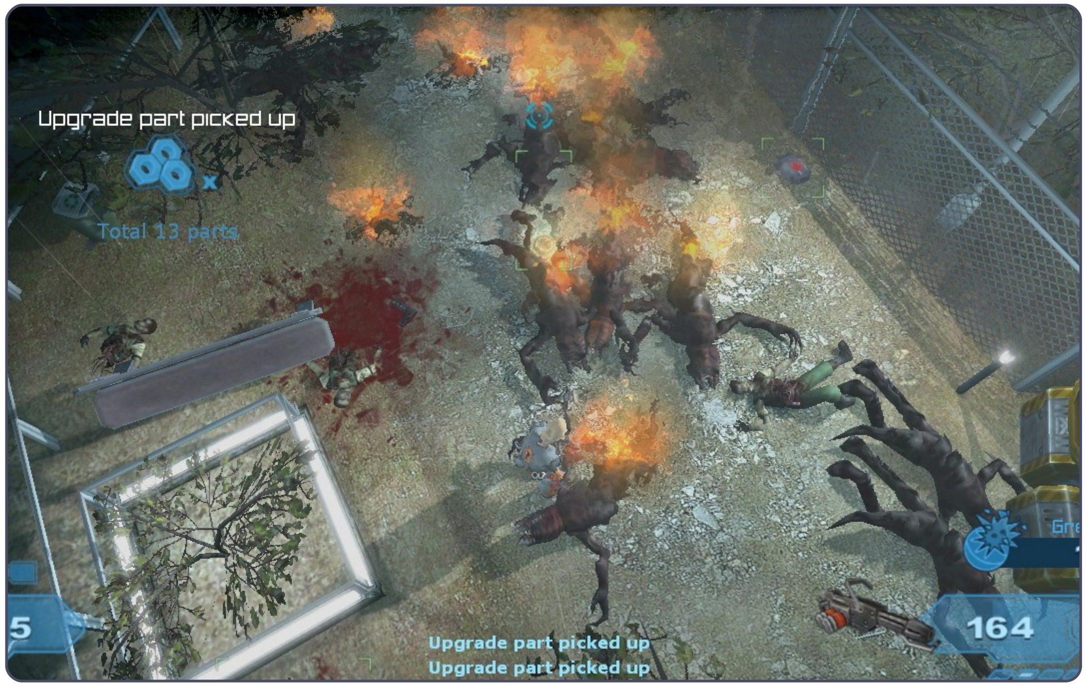
Shadowgrounds and Shadowgrounds Survivor are both solid shooters, which are slightly different to traditional action titles, thanks to its horror atmosphere and viewpoint. The story is very plain, but plenty of cut scenes and objects can be found throughout the game - which help to add some background to the main story. Missions are enjoyable, though repetitive, and bring nothing new to this shooter genre. The graphics and sound are solid throughout, and improve the atmosphere of the game. Co-Op play is really lacking, and should have been a crucial feature of this title. The port of this title to Linux is poor, and requires a good system to run this game well. Overall, it is an enjoyable shooter, but falls short in many areas.

Score: 7/10

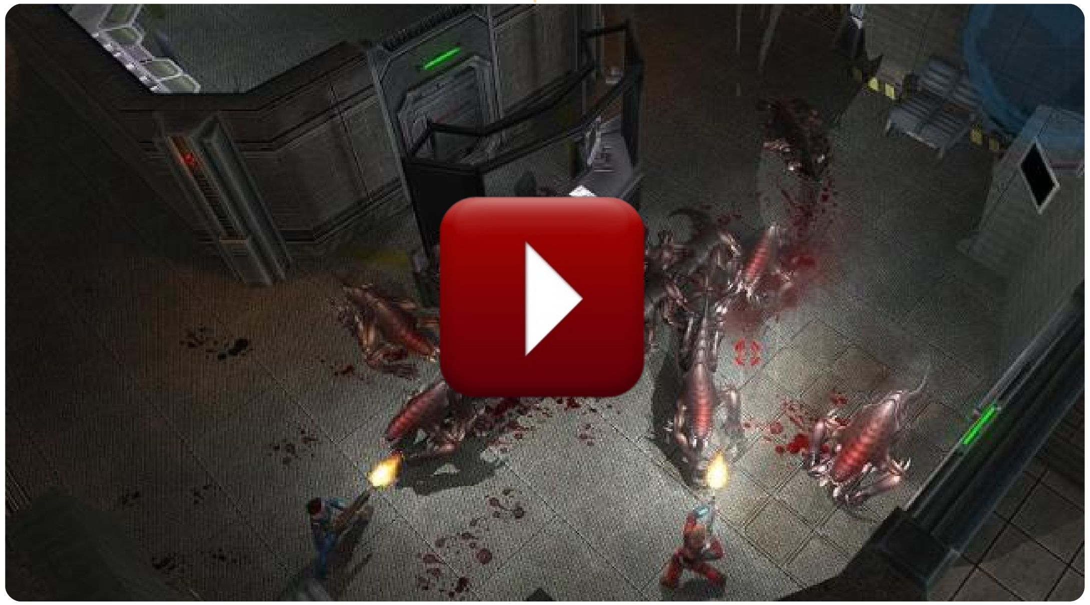
Good:
• Interesting Story, with story items to pick up during game
• Solid Gameplay
• Good Atmosphere
• Enjoyable Survival Mode (Shadowgrounds Survivor)

Bad:
• Co-Op not fully implemented (Shadowgrounds)
• Becomes dull & repetitive over time
• Poor performance on Linux

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!



Trailer:
http://www.youtube.com/watch?v=MhRedeAOWxE



contents ^

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 have always edited the grub list, but, in 10.04, if I do, and remove say, four of the repeated entries, they are put back by the system on reboot.

A The best way to deal with it is to actually remove the old kernels. At boot up, note the final five characters of the oldest kernel, such as 32-31. Run Synaptic Package Manager, and search for that string. You should get half a screen of packages, with two "linux-headers" items and one "linux-image" being installed. Right-click on each of them, and select "mark for complete removal." Then click on "Apply."

After that, open Accessories/Terminal and enter this command:

```
sudo update-grub
```

The grub list should be shorter by two items.

Q I can't seem to get my sound running on my Compaq Presario CQ56 with Ubuntu 10.04.

A Follow the instructions on this page: https://wiki.ubuntu.com/Audio/InstallingLinuxAlsaDriverModules or upgrade to Ubuntu 11.04.

Q I recently upgraded to Ubuntu 11.04, and the new scroll bars drive me crazy. How can I get the fat scroll bars back?

A Run Synaptic Package Manager, Search for liboverlay and remove it. After a reboot, you should have fat scroll bars.

Q I have just installed ubuntu 11.04 on my laptop. Everything is great, but I don't get any sound.

A Have a look at https://wiki.ubuntu.com/DebuggingSoundProblems. Odds are, your sound is muted somewhere.

Q I have installed and run Truecrypt, but it doesn't seem to encrypt files.

A When you run Truecrypt, it can create a file which is a Truecrypt "volume," and unmount or mount Truecrypt volumes. Then you can paste files in and out of the volume. When you mount a volume, you have to provide the password you used to create it, and then the contents are displayed as openly as if they were in a regular folder. However, you can upload a volume to an online storage site, and be confident that the contents are safe from casual browsing. (If the American NSA wants to see your files, all bets are off.) Then a buddy can download the volume, provide the password you have given him, and access the files.

If you want to encrypt just a single file, it is probably easier to use the Nautilus file manager. Highlight the file, right-click, and select "Compress." A window will pop up. Enter a file name, and select "7z" as the file type. Click on "Other options," and you can specify a password, and select "Encrypt the file list too." Click "Create," and you are done.

Q Let's say I've got one .avi video with a film with its original audio language (e.g. English) and another .avi with the same film but with a second audio language (e.g. re-dubbed in Italian). Is it possible to get only one .avi video with the film and the two audio languages selectable?

A This command will do it:

```
ffmpeg -i input -vcodec copy
-acodec copy output.mkv -
newaudio -i input2 -acodec
copy
```

Q I installed Ubuntu 11.04 on my netbook, leaving some unused space on the hard drive. I got it all set up the way I liked it, then I installed Android-x86 2.2 in the unused space. Now the boot menu includes only Android.

A (Thanks to Garvinrick4 in the Ubuntu Forums.) Boot from a LiveUSB flash drive, or a LiveCD if your computer has a CD drive. Open Accessories/Terminal and enter these commands:

```
sudo fdisk -l
```

(Enter your password when prompted. It should show you the storage devices, and your hard drive is probably /dev/sda. If it's not, modify the next two commands.)

```
sudo mount /dev/sda1 /mnt

sudo grub-install --root-
directory=/mnt /dev/sda

sudo umount /mnt

sudo reboot
```

Q I installed Google Chromium browser, didn't like it, so I removed it. Now when I click on a link in Evolution 2.30.3, I get a dialogue box telling me: Could not open link. Failed to execute child process "/usr/bin/chromium-browser" (no such file or directory)

A Start Firefox. Click on Edit/Preferences. Select the Advanced tab. Near the bottom, is "Always check to see if Firefox is the default browser on startup". Click the "Check Now" button. Select "yes."

Q I am an accountant. My children need to get into my computer to do homework, but I want to block spreadsheet programs to protect my work content.

A You can't block them from running programs, but you can block them from being able to access your files. Set up a non-admin userid for the kids, then in a terminal run:

```
chmod 750 /home/yourusername
```

Make sure you have a strong password they don't know!

Q My brother was messing with something to do with the look and feel of Unity, and now I have a crazy drop shadow on my mouse cursor. I hate where the drop shadow is placed and want it back to its default position.

A I found it. It's in the nVidia settings of all places.

Q I want to know how to preview sites in progress locally while using Kompozer.

A Install LAMP (in addition to Linux: Apache, Mysql and PHP), a full web server. The location of the site is /var/www. Copy your php and html files from Kompozer (javascript+images+other web content) there. In the browser you can access it via 'localhost' or 127.0.0.1.

Tips and Techniques

How hot, version 2.0

In Issue 43, I revealed one of my hang-ups: I want to know how hot things are, dammit! Then Unity arrived (for me, as a testing environment, not my production system), and applets were gone, apparently. Conky to the rescue!

Easier said than done. If someone would like to propose a "Top 5 Conky Tutorials," I think it would be a great addition to Full Circle Magazine. It was easy to find instructions on how to change the border and colour, but I honestly don't give a rat's patootie about those things. Eventually, Google led me to some useful information, but it wasn't easy. I also grabbed the official Conky manual and pasted it into a text file for offline perusal.

I installed lm-sensors before I

contents ^

got into this.

I have included my .conkyrc file. Everything up to the word "TEXT" was simply cribbed from some web site, and seemed to be OK. Displaying the uptime and the kernel version were from the same source, and they struck me as OK. Then we get into the meat of the matter. "Hwmon temp 1" turned out to be the chipset temperature, mostly based on trial and error. I had hoped there would be other "temp" variables, but it was not to be.

"Hddtemp" is the temperature reported by the hard drive. To get this, you must install hddtemp, and run it as a daemon:

hddtemp -d /dev/sda (or the name of your hard drive)

I have an Nvidia video card, and I installed a proprietary driver, and apparently that was all I needed for "nvidia temp" to work.

For the CPU temperature, a whole lot of piping was needed. "Sensors" is part of the lm-sensors package. "Cut" extracts just the desired information, and "sed" formats it.

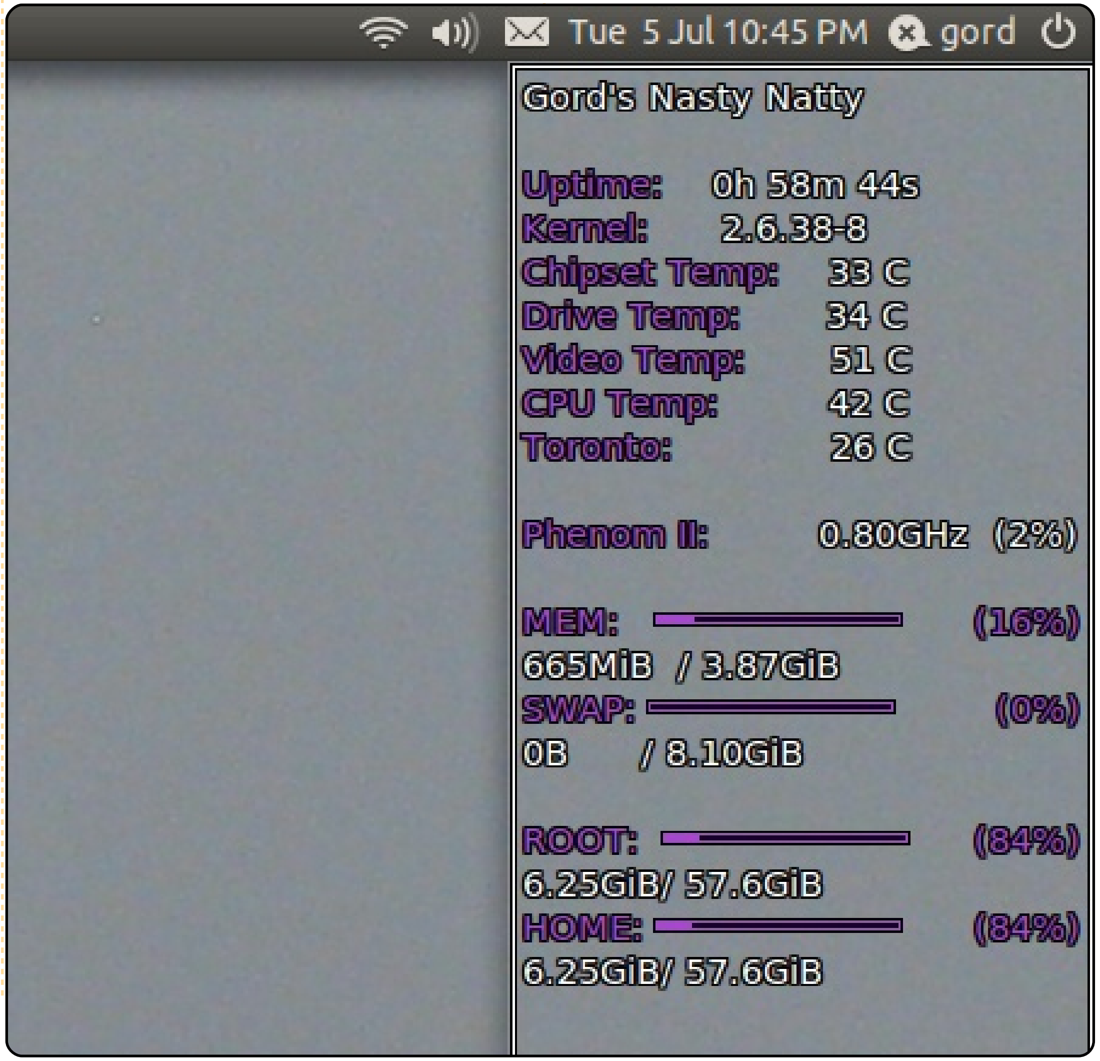
My temperature fetish is not limited to my computer, I also want to know the temperature outside. If you Google conkyforecast, you will find a place to download what turns out to be a repository name, which is added to Synaptic or Software Center. Then you can install the actual conkyforecast program. Also sign up as a partner at weather.com, if you are in North America. (Sorry, I have no idea what to do if you are outside North America. Letters please!) You will get a partner id and licence key. Create a .conkyForecast.config file in your root folder, following the pattern I have included, but with your own partner id, licence key, and default location. (Partners can display the weather on their web sites, but we're not actually going to do that.)

Again, I piped the output from conkyForecast into Cut, because the raw output included an ugly "A" with an accent.

The other items in my conky config are pretty boring: CPU frequency and utilization, memory and swap usage, and disk space usage.

The full text for the Conky discussed here (and shown below) can be seen at: http://pastebin.com/hSQwBPpT

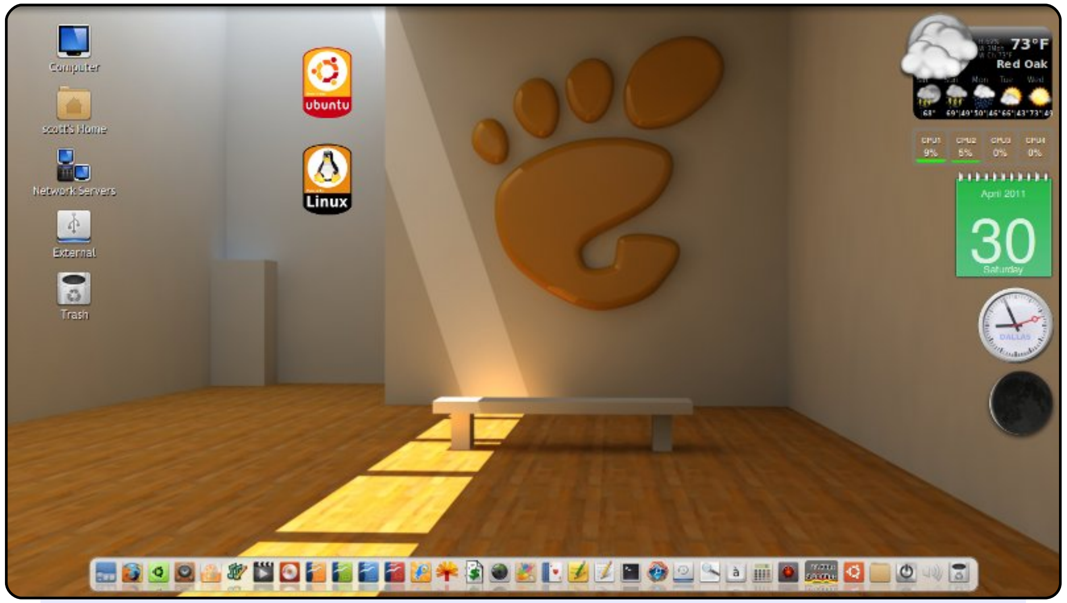
```
.conkyForecast.config

CACHE_FOLDERPATH = /tmp/
CONNECTION_TIMEOUT = 5
EXPIRY_MINUTES = 30
TIME_FORMAT = %H:%M
DATE_FORMAT = %Y-%m-%d
XOAP_PARTNER_ID = XXXXXXXXX
XOAP_LICENCE_KEY = YYYYYYYYYY
DEFAULT_LOCATION = CAXX0504
```



```
Gord's Nasty Natty

Uptime:     0h 58m 44s
Kernel:      2.6.38-8
Chipset Temp:     33 C
Drive Temp:       34 C
Video Temp:       51 C
CPU Temp:         42 C
Toronto:          26 C

Phenom II:        0.80GHz  (2%)

MEM:                      (16%)
665MiB  / 3.87GiB
SWAP:                      (0%)
0B      / 8.10GiB

ROOT:                     (84%)
6.25GiB/ 57.6GiB
HOME:                     (84%)
6.25GiB/ 57.6GiB
```

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.

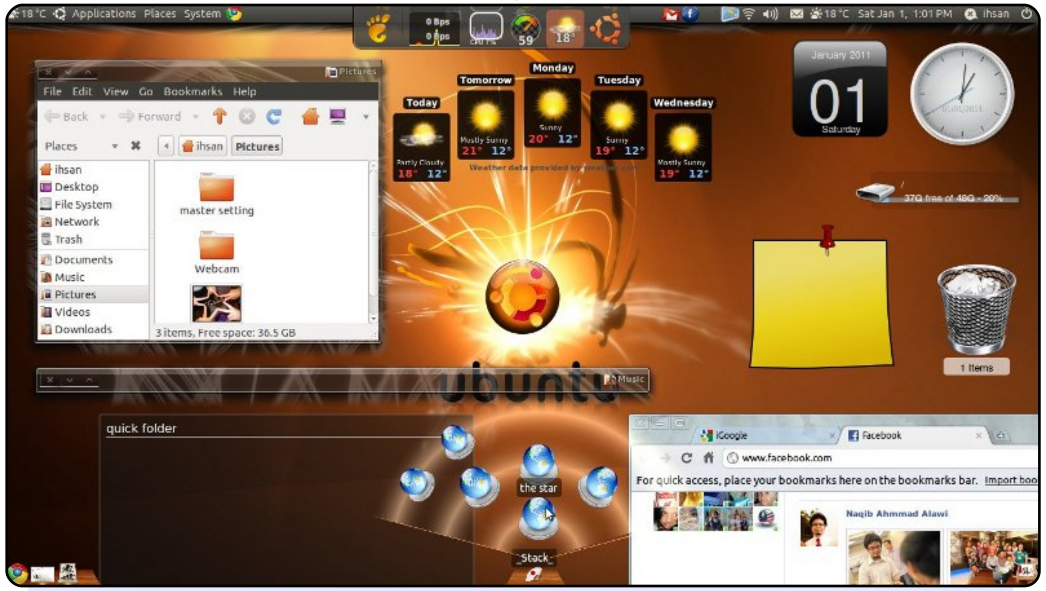


I took this screenshot before upgrading to 11.04 because I don't know what will happen with my existing desktop configuration since 11.04 will use Unity by default.

The dock at the bottom of the screen is AWN, and I have various screenlets on the right side of the screen, and 2 sticker screenlets toward the left side of the screen - to try and make them look as though they are hanging on the wall.

My computer is a Toshiba Satellite L655-S5157, with an Intel i3 processor, 4 GB of RAM (even though I'm running 32-bit Ubuntu to help ensure that device drivers work with my hardware - I still had to do some tweaking to get the wireless adapter to work and the headphones jack to work), and I upgraded to a 750 GB hard drive.

Scott M. Keeth



I took this picture some time ago. It's not really my idea of a perfect desktop, but I had fun doing it.

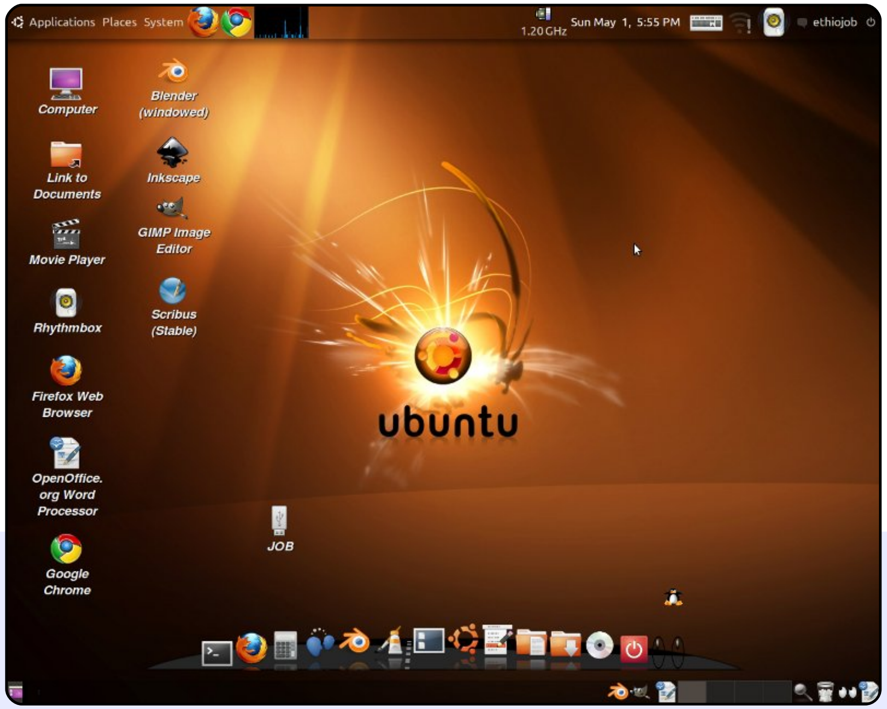
Software used: emerald theme manager, compiz, AWN, screenlets (with quick folder applets intalled), and, not to be forgotten, Cairo Dock, all running on Ubuntu 10.10, with GNOME 2.32.

Cairo Dock is heavy, and you need to have a good computer running it or you will end up having a turtle system.

Computer hardware specification: Compaq presario CQ42 203AU, RAM 3GB, graphic card ATI radeon 1GB, hard disk 320GB.

Ihsan Jaffar

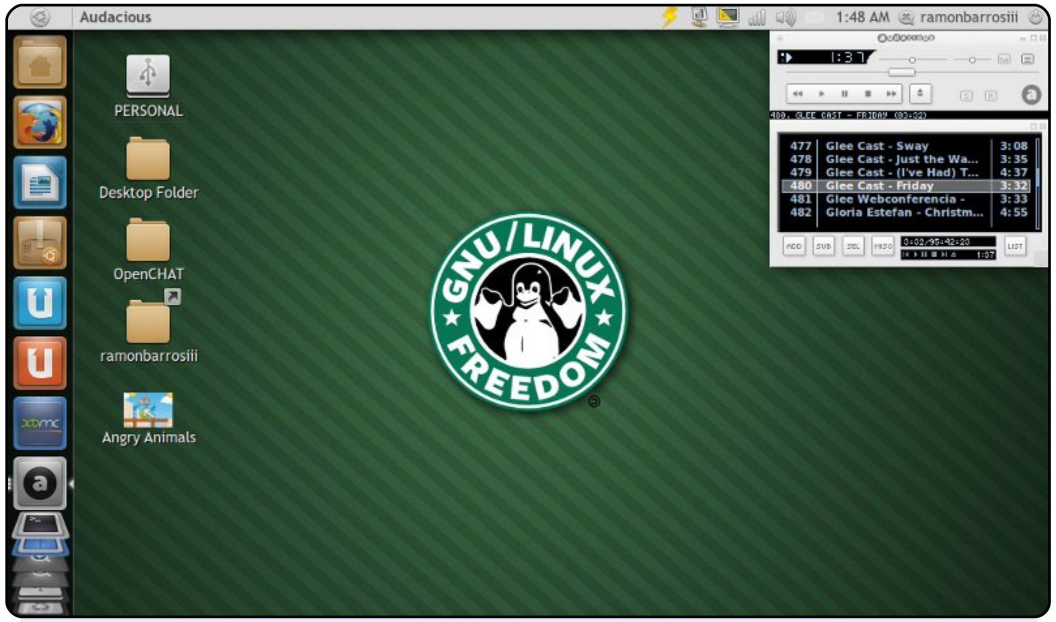
contents ^



This is my Ubuntu 10.10 desktop. I like this setup because of its simplicity and yet classic and sophisticated feel. I downloaded the wallpaper from: http://linuxhub.net/2010/01/top-35-handpicked-ubuntu-wallpapers/, used the theme Dust, adjusted the screen resolution to 1280 x 1024, and set the panels to transparent (right-click > properties...). Below that is the Cairo dock. My PC specs are:

- Dual-core 2.5 GHz
- 2GB RAM
- Integrated Graphics 256 MB

Eyob Fitwi



477 Glee Cast - Sway 3:08
478 Glee Cast - Just the Wa... 3:35
479 Glee Cast - (I've Had) T... 4:37
480 Glee Cast - Friday 3:32
481 Glee Webconferencia - 3:33
482 Gloria Estefan - Christm... 4:55

Apropos of "The Netbook is Not Dead Yet" by Allan J. Smithie in issue #49 of Full Circle, my desktop is running on an Asus EEE-PC 900E, and it runs wonderful on it. Listening to music using either Audacious (right hand corner) or Banshee Media Player, the default player, is just perfect. The wallpaper is from:

http://lalitpatanpur.deviantart.com/art/GNU-Linux-quot-Starbuck-s-quot-logo-141284973

I'm always on deviantart looking for Ubuntu wallpapers - wonderful site for it. The skin is elementary, and the icon is Faenza-Dark set.

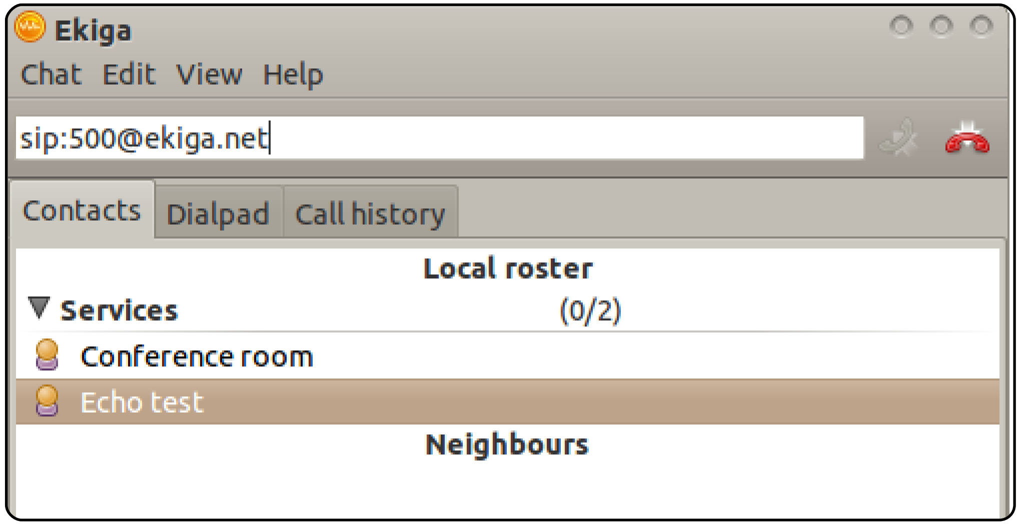
Ramon Barros

Ekiga

Homepage: http://ekiga.org/

Ekiga, originally written by Damien Sandras as a master's thesis, is one of the most popular open source softphones, particularly after the acquisition and subsequent shutdown of Gizmo5 by Google. One of the main reasons is its ease of use. Most VoIP clients, for example, require you to sign up with an external service. While you can use a third party server, Ekiga offers a built-in service, helping new users feel right at home. But don't be fooled into thinking Ekiga is only for the novice user; the application sports many advanced features, including support for a laundry list of codecs and LDAP address-book lookup. Users thinking about moving to Ekiga full-time will also be happy to discover Ekiga Call Out, which allows users to call "real" phone numbers for cheap rates.

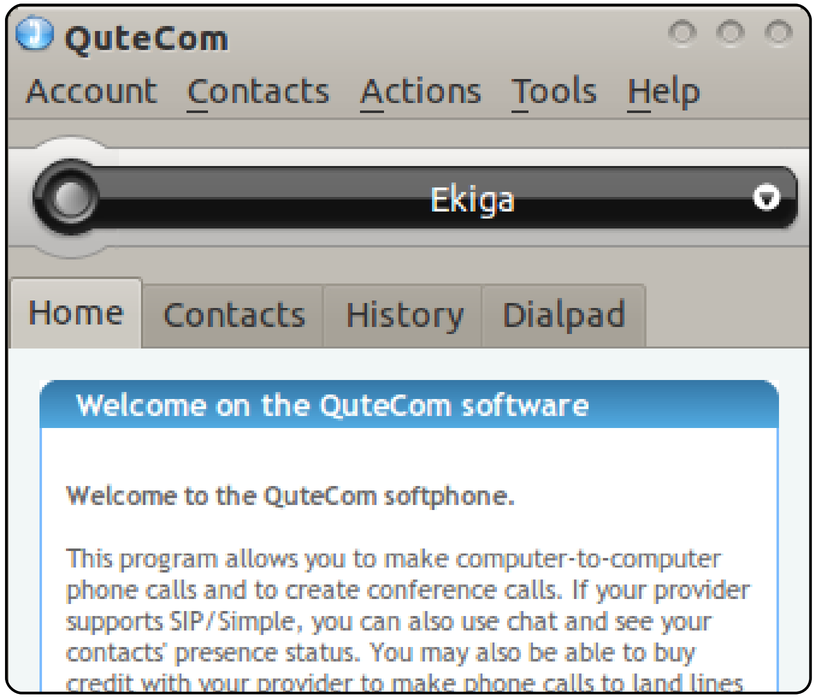
To install Ekiga, use the ***ekiga*** package in the universe repositories.



QuteCom

QuteCom (formerly known as WengoPhone and run by the French VoIP service Wengo) is another highly popular SIP client. Like Ekiga, it supports voice and video chatting. Where it excels is its support for third party protocols. Its developers have implemented support for the libpurple library, the library powering the popular cross-platform program Pidgin. As a result, QuteCom users can chat with MSN, AIM, ICQ Yahoo, Jabber, Facebook, MySpace, and Skype users (though support for Skype is buggy and has questionable legal ramifications).

To install QuteCom, use the ***qutecom*** package in the universe repositories.


QuteCom
Account Contacts Actions Tools Help
Ekiga
Home Contacts History Dialpad
Welcome on the QuteCom software

Welcome to the QuteCom softphone.

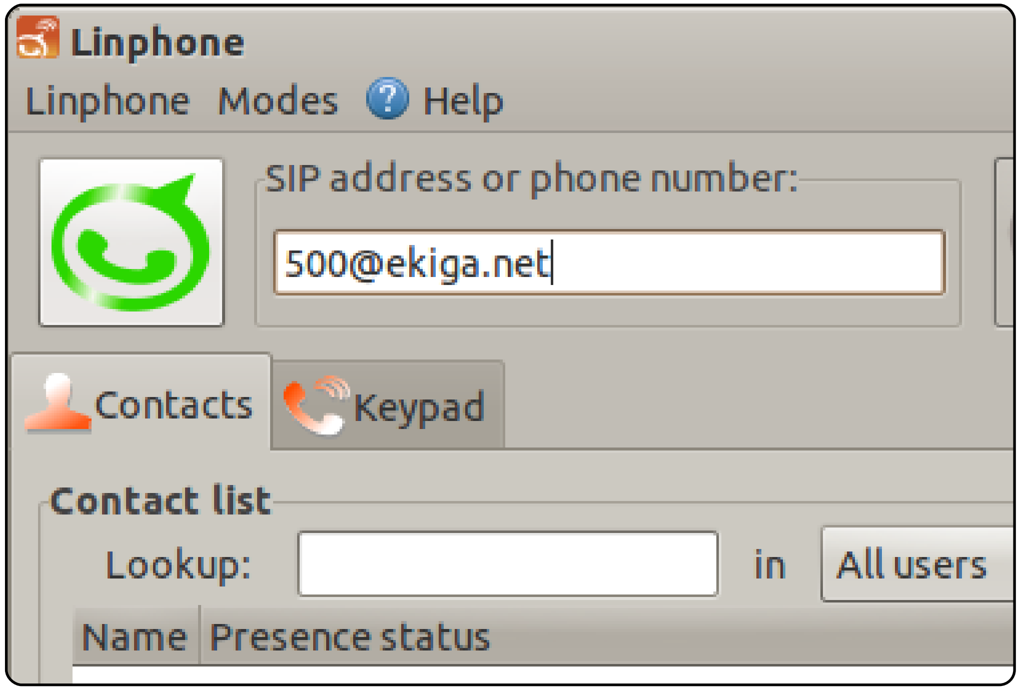
This program allows you to make computer-to-computer phone calls and to create conference calls. If your provider supports SIP/Simple, you can also use chat and see your contacts' presence status. You may also be able to buy credit with your provider to make phone calls to land lines

Linphone

Homepage: http://www.linphone.org/

If you want a slightly more configurable SIP client (with a much less user-friendly interface), check out Linphone. It has a bevy of advanced configuration settings, including IPv6/IPv4 switching, manual RTP/UDP ports, maximum transmission unit configuration, and so on. Additionally, it's cross-platform - you can use the app on Android, Blackberry, or your iPhone, a nice feature if you want a uniform interface. Finally, for you terminal junkies, there's a built-in command line interface.

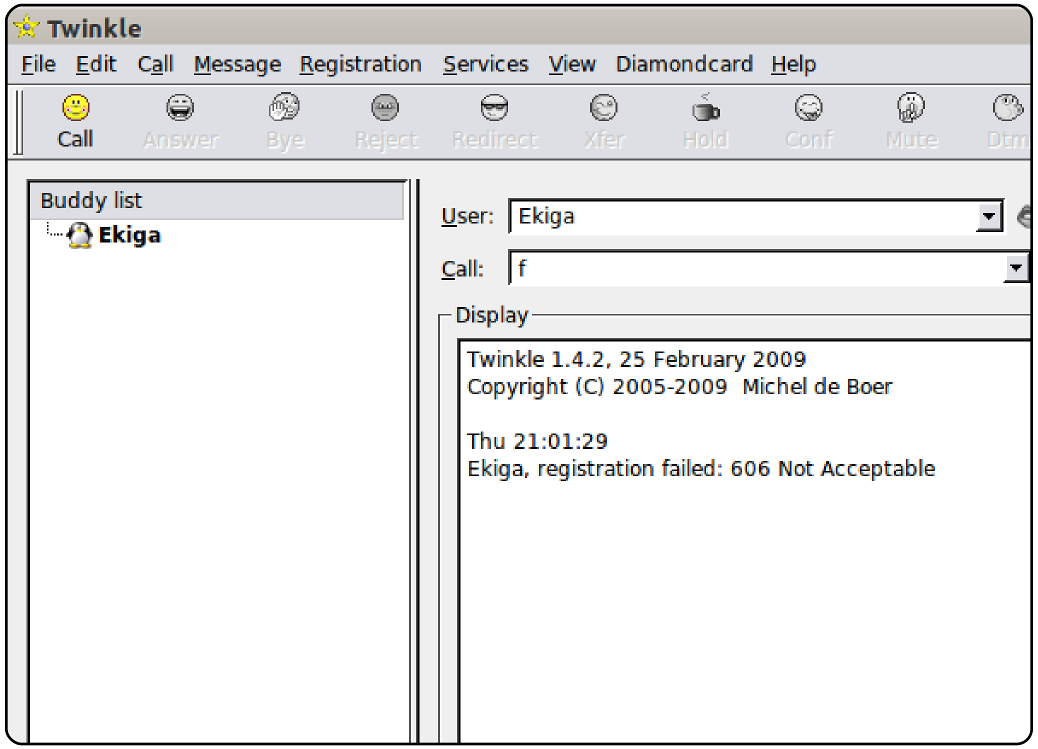
To install Linphone, use the Ubuntu package at the official download page.



Twinkle

Homepage: http://www.twinklephone.com/

Twinkle has always been my favorite KDE SIP client. To start, it's incredibly user friendly. Its wizard interface for setting up accounts includes built-in support for FreeWorld Dialup, sipgate, SIPPhone (though SIPPhone, run by Gizmo5, is currently defunct), and Diamondcard, which lets you make calls to landlines and other "real" phones. There's also lots of KDE integration; of particular use is the KAddressBook integration (though you can use the built-in address book if you don't use KDE). Finally, for the scripters and coders of the world, Twinkle offers event scripts. You can configure various Bash scripts to execute when certain events (incoming call, outgoing call, call released, etc.) are triggered.



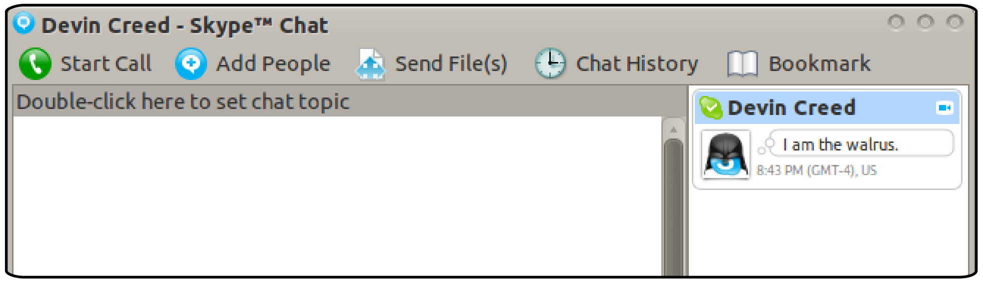
```
Twinkle 1.4.2, 25 February 2009
Copyright (C) 2005-2009  Michel de Boer

Thu 21:01:29
Ekiga, registration failed: 606 Not Acceptable
```

Skype

Homepage: http://www.skype.com/

No list of VoIP clients would be complete without Skype, the grandfather of softphones, recently acquired by Microsoft for $8.5 billion. Unfortunately, even before the Microsoft acquisition, Skype's Linux support lagged. While both Windows and Mac users have access to 5.x builds, Linux users are forced to use 2.2. That means quite a few features, including group video, are missing. You'll also be stuck with a slightly dated interface - though, if you've seen the more recent iterations of the Windows interface, that might not be a bad thing. Most unfortunate of all, Skype uses its own proprietary protocol - you have to have a Skype account, and you can't officially use any third party clients to connect to it.

To install Skype, download the .deb package for Ubuntu from the official homepage.



I am the walrus.
8:43 PM (GMT-4), US

Top 5 - THE END

Unfortunately Andrew no longer has the time to continue writing the Top 5 and is leaving FCM. It's been a joy to work with him for the past four years and I hope you'll all join me in wishing him the best of luck in his endeavours.

ubuntu-uk
local community support team

The Ubuntu UK podcast is presented by members of the United Kingdom's Ubuntu Linux community.

We aim is to provide current, topical information about, and for, Ubuntu Linux users the world over. We cover all aspects of Ubuntu Linux and Free Software, and appeal to everyone from the newest user to the oldest coder, from the command line to the latest GUI.

Because the show is produced by the Ubuntu UK community, the podcast is covered by the Ubuntu Code of Conduct and is therefore suitable for all ages.

http://podcast.ubuntu-uk.org/

ubuntu uk podcast

Download

Available in MP3/OGG format in Miro or iTunes, or listen to it directly on the site.

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
David Haas
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 #52:
Sunday 07th August 2011.

Release date for issue #52:
Friday 26th August 2011.

contents ^