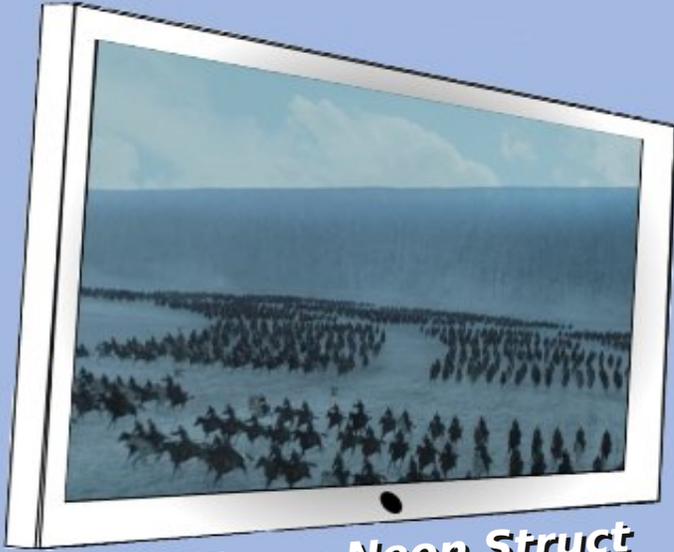


# The **PCLinuxOS** magazine

Volume 101

June, 2015



*Game Zone: Neon Struct*

**Monitor  
Your  
System  
With  
Conky,  
Part I**

**Monitor  
Gamma  
Correction  
Made  
Simple**



**Inkscape  
Tutorial:  
Abstract  
Wallpaper**

**Downsizing: Moving From KDE To Mate**

**Leveraging MPD  
(Music Player  
Daemon)**



**HTPC: Getting  
Started With Kodi**

**And More Inside ...**

# Table Of Contents

- 3 Welcome From The Chief Editor**
- 4 Downsizing: Moving From KDE To Mate**
- 9 Screenshot Showcase**
- 10 HTPC: Getting Started With Kodi**
- 33 Screenshot Showcase**
- 34 PCLinuxOS Recipe Corner**
- 35 ms\_meme's Nook: PCLOS Forever**
- 36 Monitor Your System With Conky, Part One**
- 50 Inkscape Tutorial: Abstract Wallpaper**
- 52 Screenshot Showcase**
- 53 Tip Top Tips: Samsung Printer Driver Installer**
- 54 Screenshot Showcase**
- 55 Leveraging MPD (Music Player Daemon)**
- 63 PCLinuxOS Family Member Spotlight: wayne1932**
- 64 PhotoPrint: Application For Fast Photo Layout And Printing**
- 65 YAT: Yet Another Testimonial**
- 66 RetroShare Tutorial**
- 70 Screenshot Showcase**
- 71 Game Zone: Neon Struct**
- 72 Screenshot Showcase**
- 73 Monitor Gamma Correction Made Simple**
- 76 PCLinuxOS Puzzled Partitions**
- 79 More Screenshot Showcase**

## The PCLinuxOS magazine

The PCLinuxOS name, logo and colors are the trademark of Texstar.

The PCLinuxOS Magazine is a monthly online publication containing PCLinuxOS-related materials. It is published primarily for members of the PCLinuxOS community. The magazine staff is comprised of volunteers from the PCLinuxOS community.

Visit us online at <http://www.pclosmag.com>

This release was made possible by the following volunteers:

Chief Editor: Paul Arnote (parnote)  
Assistant Editor: Meemaw  
Artwork: ms\_meme, Meemaw  
Magazine Layout: Paul Arnote, Meemaw, ms\_meme  
HTML Layout: YouCanToo

Staff:

ms_meme	loudog
Meemaw	YouCanToo
Gary L. Ratliff, Sr.	Pete Kelly
Daniel Meiß-Wilhelm	Antonis Komis
daiashi	Khadis Thok
Smileeb	

Contributors:

Odin	DanielBoone
------	-------------

The PCLinuxOS Magazine is released under the Creative Commons Attribution-NonCommercial-Share-Alike 3.0 Unported license. Some rights are reserved.  
Copyright © 2015.



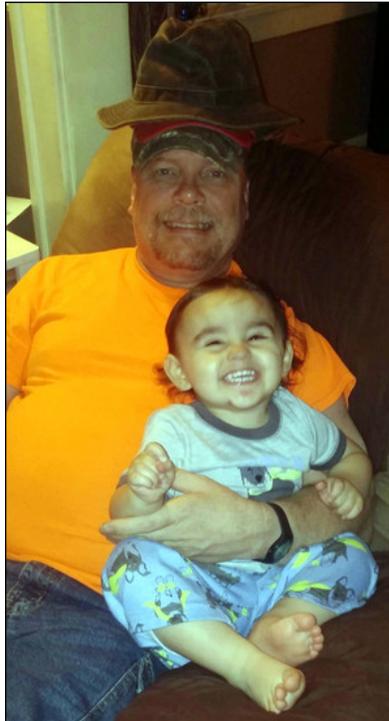
# Welcome From The Chief Editor

My son, Ryan, who will be 22 months old in June, thinks that whenever he sees a hat lying around, you should be wearing it. And, it makes little difference to him if you already have another hat on. He expects you to wear ALL the hats that are around. All at the same time.

I'll give him props on knowing where hats go – on your head. He even has his own hat. Not that he wears it, mind you, but it is his hat. Good luck trying to get him to wear his hat, as it most likely won't happen. But, if his hat is among the mix of other hats around, he thinks you should wear his hat, too. There is no regard in his mind if it fits or not. He just expects you to wear ALL the hats that are around. All at the same time.

When you stop and think about it, Ryan wanting me to wear all the hats that are laying around is a lot like what's expected of us in life. Lots of us are expected to wear many different hats in life, sometimes all at once.

As a parent, I'm his protector, his provider, his moral compass, his disciplinarian, his educator, his friend, and his Mr. Fix-It. I'm expected to provide for his safety, kiss his boo-boos when they happen, and set the boundaries while enabling him the liberties to explore his world. I make sure he has food when he's hungry, drink when he's thirsty, and a clean and dry diaper. Quite often, I have to do all or many of these things – all at the same time.



In my job at the hospital, I'm a listener and talker, all at the same time. I provide compassionate care to my patients, while at the same time, putting on my clown nose (figuratively) and/or being a reference resource. I'm an educator and a trusted friend. I'm diplomatic when I need to be, and sternly authoritative at other times. I can move – directly and immediately – from the delivery room where I may have just helped a 26 week gestation premature baby gain a foothold on life, to taking care of someone's dying 90 year old grandmother who is losing their foothold on life.

Just as in the picture above where Ryan has placed four different hats on my head, I'm sure that each of us can relate to having to wear multiple hats, all at the same time, at one time or another. And

honestly, I don't think there's a single person reading this that isn't expected to wear multiple hats at the same time, every day. You may not recognize it, but it happens to us all, every single day. If you don't see it, it just means that you haven't taken the time for that introspection.

Isn't it funny how a 22 month old infant can bring to light something that many of us rarely give a second thought to? We may not always wear those multiple hats gracefully every day, but wear them we do – whether we like it or not.

Until next month, I bid you peace, prosperity, serenity and happiness.

## Disclaimer

1. All the contents of The PCLinuxOS Magazine are only for general information and/or use. Such contents do not constitute advice and should not be relied upon in making (or refraining from making) any decision. Any specific advice or replies to queries in any part of the magazine is/are the person opinion of such experts/consultants/persons and are not subscribed to by The PCLinuxOS Magazine.
2. The information in The PCLinuxOS Magazine is provided on an "AS IS" basis, and all warranties, expressed or implied of any kind, regarding any matter pertaining to any information, advice or replies are disclaimed and excluded.
3. The PCLinuxOS Magazine and its associates shall not be liable, at any time, for damages (including, but not limited to, without limitation, damages of any kind) arising in contract, tort or otherwise, from the use of or inability to use the magazine, or any of its contents, or from any action taken (or refrained from being taken) as a result of using the magazine or any such contents or for any failure of performance, error, omission, interruption, deletion, defect, delay in operation or transmission, computer virus, communications line failure, theft or destruction or unauthorized access to, alteration of, or use of information contained on the magazine.
4. No representations, warranties or guarantees whatsoever are made as to the accuracy, adequacy, reliability, completeness, suitability, or applicability of the information to a particular situation. All trademarks are the property of their respective owners.
5. Certain links on the magazine lead to resources located on servers maintained by third parties over whom The PCLinuxOS Magazine has no control or connection, business or otherwise. These sites are external to The PCLinuxOS Magazine and by visiting these, you are doing so of your own accord and assume all responsibility and liability for such action.

### Material Submitted by Users

A majority of sections in the magazine contain materials submitted by users. The PCLinuxOS Magazine accepts no responsibility for the content, accuracy, conformity to applicable laws of such material.

### Entire Agreement

These terms constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes and replaces all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter.

# Downsizing: Moving From KDE To Mate

by Peter Kelly (critter)

In last month's magazine I wrote about changing from KDE to the Mate desktop environment. This involved my efforts to apply some of the more useful features of KDE to Mate using a utility called `xdotool`. One of the features in KDE that I particularly liked was the ability to have an application open at a pre-determined location and of a particular size, maybe also on a particular workspace.

Mate is a desktop environment of which the window manager is only one element. It is possible to use a different, more powerful window manager than the default Marco, but the 'up-market' managers tend to be less economical on resources, which would defeat the object of my moving to Mate. Anyway, Marco is a good, lightweight window manager, with a very respectable heritage from Gnome's Metacity, around which Mate has been designed. Little Marco just needs a little help, is all.

I had achieved my goal of getting this particular feature to work by using `xdotool`. However, the results were less than satisfactory, briefly displaying the window before moving and re-sizing it to the desired parameters. I thought that there must be a better way to do this and so, once again, I turned to the internet.

What I was looking for was something that would work with the Marco window manager (Wikipedia tells me that marco is Spanish for frame, which I suppose makes sense for a window manager\*\*). Whichever tool I used should also be available from the PCLinuxOS repositories.

\*\*For those Mate users who may have wondered about the strange names used for some of the Mate applications:

When the Gnome desktop moved from version 2 to 3, many users, including Linus Torvalds, did not like the direction taken by the new system and started to look for alternatives. One of these disgruntled users, an Argentinian Arch user named Perberos, decided to take a different approach and forked Gnome 2 to a new desktop environment (DE) he named Mate, after the popular South American beverage.

The Mate utilities that he forked from Gnome had to be re-named and for this he chose to use names from his native Spanish. Wikipedia translates them thus:

Caja (box) – File manager (from Nautilus)

Pluma (quill) – Text editor (from Gedit)

Eye of MATE– Image viewer (from Eye of GNOME)

Atril (lectern) – Document viewer (from Evince)

Engrampa (staple) – Archive manager (from File Roller)

Mate Terminal – Terminal emulator (from GNOME Terminal)

Marco (frame) – Window manager (from Metacity)

Mozo (waiter) – Menu item editor (from Alacarte)

## wmctrl

The first tool I looked at was `wmctrl`, which can be a useful utility. But, it steadfastly refused to recognize some applications, most notably LibreOffice and Firefox. Perhaps this is related to the fact that the version in the PCLinuxOS repositories is 1.07 and a visit to the applications website showed this to have been released in 2005. A version 1.08 was mentioned, but since this was also released in 2005, there is still ten years of window management development not addressed. This inability to recognize all windows rendered it unsuitable for the particular task at hand.

## devilspie



The next little gem that I stumbled upon carries the unlikely name of `devilspie`. This is another tool that seems to be no longer maintained, 2007 as near as I can tell, so again quite old. There is a more recent fork from this named `devilspie2`, but that is not in the repositories and so that one is off the list. From the

website of the developer, Ross Burton, comes this quote:

*"A window-matching utility, inspired by Sawfish's "Matched Windows" option and the lack of the functionality in Metacity"*

This looks more promising, despite its age.

Devilspie works by looking at the windows managed by the window manager, and if they match criteria listed in a pre-written rule, then the actions described in the rule are applied to that window. The rules are text files that end with the extension '.ds' and are stored in a folder in your home directory named '.devilspie' which you will have to create (don't forget the leading dot). The rules follow a format known as 's-expressions' developed for the lisp programming language, but that is nothing to be alarmed about. The format is quite straight forward.

Documentation is rather poor, but a good reference to the available functions, commands etc. can be found here <http://www.foosel.org/linux/devilspie>. This website also provides more links for information.

To force my pluma text editor to always open on the second virtual desktop, in the top-left corner of the screen and 700 pixels wide by 500 pixels high, I have to create a file containing a rule to apply to pluma windows with the following text:

```
(if
(is (application_name) "Pluma")
(begin
(set_workspace 2)
(geometry "700x500+0+0")
)
)
```

The filename is unimportant, but must end in **.ds**, and must be saved to a directory in your home directory named **.devilspie**.

So

`/home/me/.devilspie/pluma.ds`

is how I saved my file. The file first looks for a window that has the name "Pluma" and if found it applies the two rules following the begin statement. Workspaces are numbered from one here, in contrast to xdotool which begins numbering from zero. This method works better than my previous attempt, but does still flash briefly on the screen before the rules are applied. So, for this use, the search goes on. But for now, I am content.

Looking a little deeper into the abilities of devilspie, I can see several ways to utilize this tool. I use LibreOffice quite frequently, and when I launch the word processor, I like to have just the text across the window, no distracting side bars, but the height maximised to display as much of the text as possible. On the other hand, the spreadsheet is always maximised and on a separate virtual desktop. Here the Application Name is 'LibreOffice 4.4' for both applications (4.4 is the current version on my system). However, the window class though differs between applications, being 'libreoffice-writer' or 'libreoffice-calc' as appropriate. Knowing this I can now write a different set of rules for each application (See below for how to get this information).

writer.ds

```
(if
(is (window_class) "libreoffice-writer")
(begin
(geometry "1100x800+290")
(maximize_vertically)
)
)
```

The window is first resized horizontally to the desired size and vertically to an arbitrary value. It is then maximized in the vertical direction only. The geometry statement needs a little explanation. The values I have shown are used because I use two monitors, side by side, with resolutions of

290                      1100                      290



1680x1050 and 1920x1080 respectively which gives an effective resolution of 3600x1080. As I want the word processor to be 1100 pixels wide and placed centrally on the first monitor, I need an x value of  $(1680 - 1100)/2 = 290$ .

If I was using only one monitor, then I could ignore the calculation, omit the +290 offset, and simply add a center statement, which would calculate the correct position for me. Using the center statement with multiple monitors is, at best, unpredictable.

The spreadsheet, being maximized in both directions, is more straight forward.

calc.ds

```
(if
(is (window_class) "libreoffice-calc")
(begin
(set_workspace 2)
(maximize)
)
)
```

An easy way to get the information to write your rules is to open the relevant applications, and then run devilspie debug in a terminal. Alternatively, you can put a file named debug.ds into the .devilspie folder containing just the single word debug. Rename the file to just debug when you no longer

need the information. With this in place, issuing the command

## devilspie debug

Press Control + c to regain control.

Will print out the following information for each window:

Window Title, Application Name, Class and Geometry. The first three are matched by the strings window\_name, application\_name and window\_class respectively.

When trying to match windows, you have quite a lot of flexibility. Instead of using the test is

```
( is string_a string_b )
```

which is a test of equality as used in the above examples, you can use **contains**

```
( contains string_a string_b )
```

which looks for string\_b within string\_a (string\_b is a substring of string\_a).

Alternatively there is the test **matches**

```
( matches string_a pattern )
```

Where pattern is a regular expression. This greatly increases the flexibility of the search, and allows groups of windows to be matched. It is also possible to combine multiple tests using the so called boolean operators: and, or & not. The way to use these is quite simple, but may not be immediately obvious. The general format looks like this.

```
(begin # start here
(if # if, and only if
(is (some test) value) # this is true
(and # and
(is (some other test) value) # this is
also true
```

```
(not (is (another test) value)) # but this
test is
false
(begin # then, and only then
(action 1) # do this
(action 2) # and this
... # and these
) # end of second begin
) # end of and
) # end of if
) # end of first begin
```

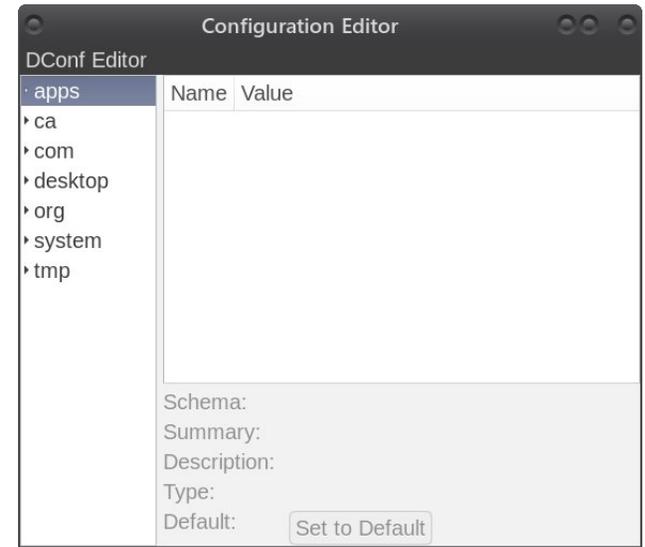
Just make sure that the parentheses are correctly closed out, and use indentation for readability. The link provided earlier will give you more information and describes the following actions that can be performed by devilspie:

**debug, print, println, str, hex, geometry, fullscreen, focus, center, maximize, maximize\_vertically, maximize\_horizontally, unmaximize, minimize, unminimize, shade, unshade, close, pin, unpin, stick, unstick, set\_workspace, set\_viewport, skip\_pager, skip\_tasklist, above, below, undecorate, wintype, opacity, spawn\_sync and spawn\_async.**

This is quite a comprehensive list that should enable me to fill in more of the gaps left by my move from KDE.

## The dconf editor

Also in last month's article I briefly mentioned the dconf editor. This tool is not specific to Mate, and is used by other desktops, such as Xfce. It is a graphical front end used to access and adjust some of the lower level system settings. Unlike most Linux configuration settings, these are not stored in a simple text file intended to be edited with a text editor. Instead, they are stored in a database as key - value pairs, and the dconf editor enables you to more easily change these values. The command line tool for doing this is gsettings. The command line



tool is not very easy to use, but the graphical editor is most welcome, albeit being a little quirky.

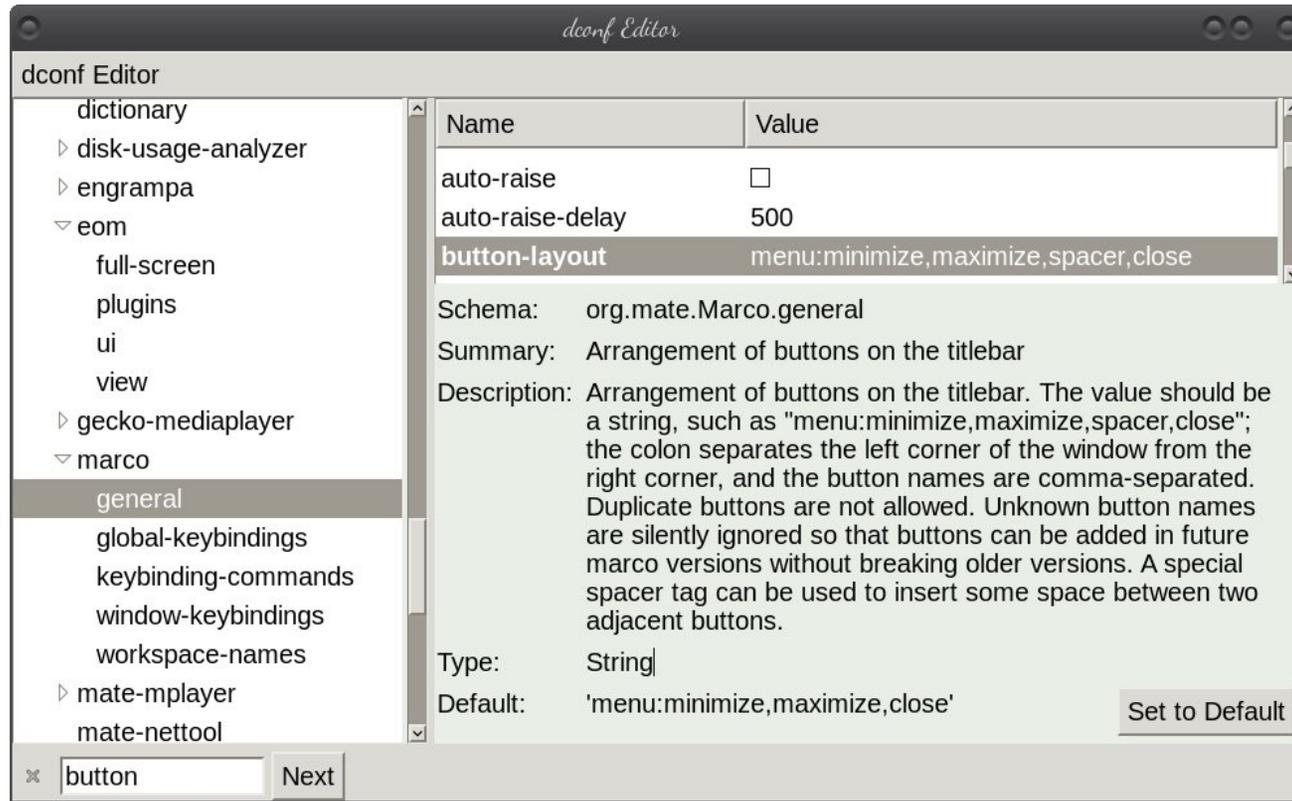
As these settings are critical to how your desktop appears and behaves, it would be prudent to make a backup of the current settings before you begin to change things that you later regret. The file to back up is `~/.config/dconf/user`.



There is no menu as such where you usually find items like open and save, but if you click on Dconf Editor in the top-left, you will be presented with a three item drop down list containing Find, About and Quit options.



Selecting the "Find" option opens a small text entry box bottom-left where you can search for an item, which it may or may not decide to find for you. Alternatively, you can navigate through the tree on



the left expanding sub-sections as required by clicking on the little triangles.

I entered the word button in to the find box and got lucky. It found what I was looking for fairly quickly. These buttons are the ones incorporated in the window title bar, where you can, for example, maximize or close the window. The dconf editor gives some helpful information about what the values for this key may be, the effect this will have and what the default value is. This button-layout key appears in several places, but the one that worked for me using Mate was under marco > general > button-layout.

I like to have the close button separated slightly from the minimize and maximize buttons to avoid accidental closures. To incorporate this feature I

have inserted the value 'spacer' in the comma separated list. You can see the effect of this in the image above. You may have to log out or even reboot to see the effects of your changes.

If your desktop environment uses the dconf editor, then it is worth taking a look at. There are many settings here that are difficult or almost impossible to adjust in any other way, and there are some that you almost certainly didn't know about. While I was exploring the org>mate>marco sections I discovered several keyboard shortcuts had been enabled that I was not aware of, and many more ready to be assigned for those who may find them useful.

One more thing that I missed from KDE was the ability to open man pages in konqueror via a kioslave. This just means that I got a much more civilized presentation of the man pages content which, in its usual terminal style output, is pretty dire.

Mate has a help file browser called yelp which, according to the documentation, can also be used to display man and info files. Unfortunately, this feature does not seem to be available in the current PCLinuxOS Mate installation.

To work around this, I called on a little bash function I discovered way back when and created an alias to it in my ~/.bashrc file. This will also accept section numbers as in `man 8 lsblk`.

This is the function as I implemented it in my ~/.bashrc file:

```
# found this on reddit a while back
# Posted by sledgespread - thanks
# sledgespread
pdfman()
{
    TMPFILE=$(mktemp atril-
$USER.XXXXXX --tmpdir)
    man -t "$@" | ps2pdf - "$TMPFILE"
    atril "$TMPFILE"
    bash -c "sleep 2; rm $TMPFILE" &
}
```

And, of course, the alias.

```
alias man=pdfman
```

In this function, a temporary file is created and the ps2pdf utility is used to convert the man output to PDF format, which is then displayed on screen. The conversion is surprisingly quick, hardly any noticeable delay. Here, I'm using Mate's Atril PDF display utility to show the PDF files, but this can be substituted for your own favorite PDF reader.

If the manual page does not exist, then you get a blank PDF displayed and the message "No manual entry for..." is displayed in the terminal. While this can be considered a bug, I considered it not worth fixing, as the blank file tells the story. The temporary PDF file is destroyed on exiting the function.

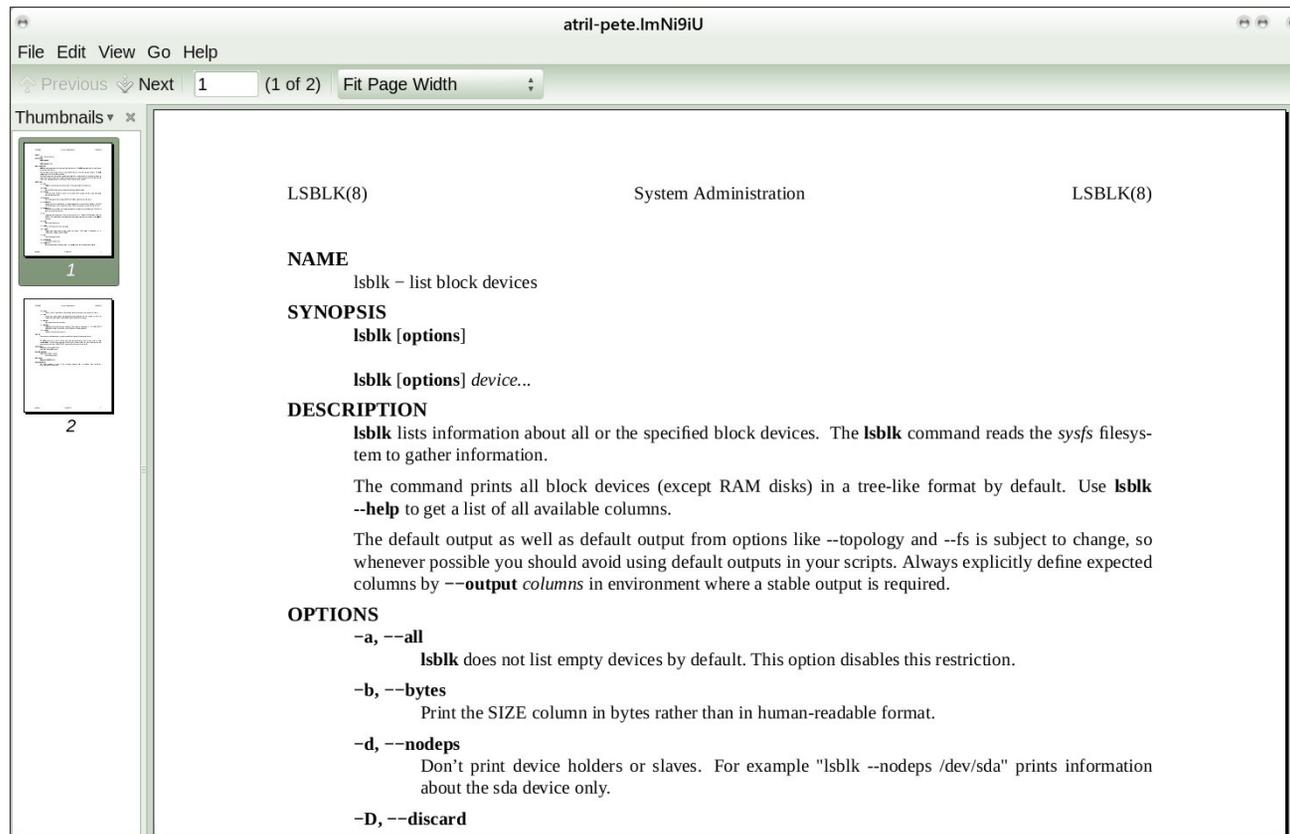
While this is not a the most efficient process, it works well and resource usage is low and only temporary. I may investigate expanding this to view info files. Well, maybe one day I will, but I really don't use info files much, since I can usually get clearer information from the internet.

The ps2pdf utility is part of the ghostscript-common package which, if not already installed, is available in the PCLinuxOS repositories.

So now I have Mate mostly set up that I can use it in the way that I used KDE. Whatever features are still missing, I now have a good idea how to set them up. Plus, I have access to some features that I never knew in KDE. My Mate setup is snappy and responsive, while demanding less from my hardware. As a bonus, should I ever decide to switch

to another desktop environment, I now know how to get the most out of it.

Linux has always been extremely configurable, enabling users to more precisely adjust it to their needs and preferences. These few tools that I have explored can give you access to even more of these features.





Support PCLinuxOS! Get Your Official

**PCLinuxOS**  
Merchandise Today!

# PCLinuxOS



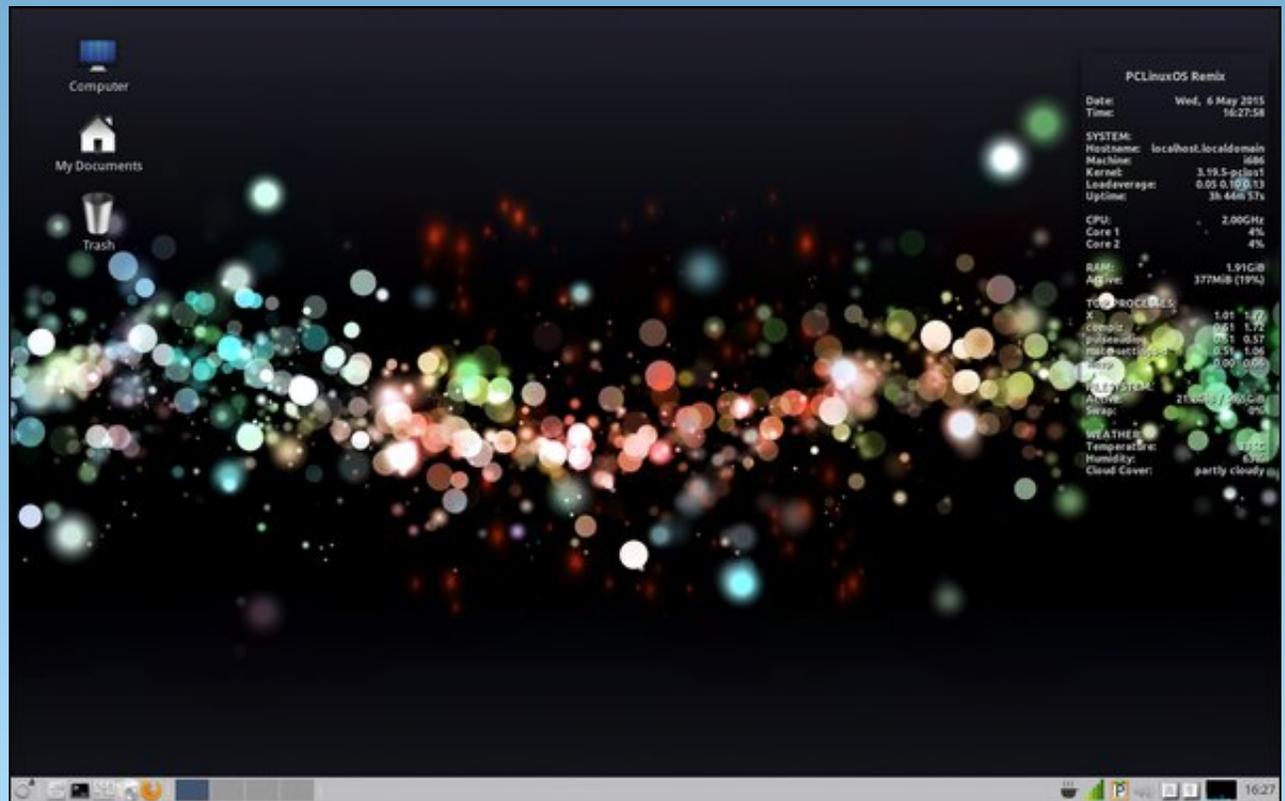
## PCLOS-Talk

Instant Messaging Server



Sign up TODAY! <http://pclostalk.pclosusers.com>

## Screenshot Showcase



Posted by zen-subz, May 6, 2015, running Mate.



# HTPC: Getting Started With Kodi

by Paul Arnote (parnote)

Kodi ... formerly called XBMC ... is an open source multimedia powerhouse of a program that helps you organize your photos, music, and videos. It was originally released in 2002 as a media center application for the original Xbox, but it no longer supports the Xbox (with the exception of a third party fork, called XBMC4Xbox). In August 2014, XBMC officially changed its name to Kodi to somewhat sever its link to the now unsupported Xbox.

It is available for Linux, Windows, BSD, Mac OS, iOS and Android platforms. Because of its open source and cross-platform nature, with its core code written in C++ (ANSI standard), modified versions of Kodi/XBMC together with a [JeOS](#) have been used as a software appliance suite or software framework in a variety of devices including smart TVs, set-top boxes, digital signage, hotel television systems, and network connected media players. [Derivative](#) applications such as MediaPortal, Plex, ToFu, Voddler, and Horizon TV have all initially been spun off from Kodi/XBMC. There are also special versions of Kodi available for specific computer platforms, such as the Raspberry Pi. Kodi is available for installation from the PCLinuxOS repository.

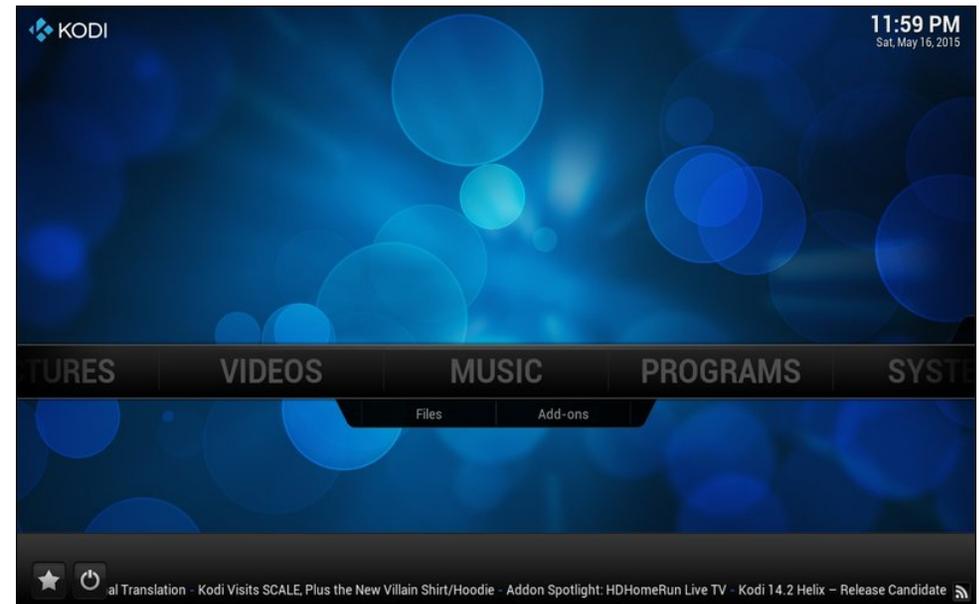
What Kodi isn't is a media server, even though it has elements of a media server built in. Some people have even successfully used Kodi as a media server. But that is not Kodi's intended role. Kodi shines when used for its intended purpose, which is that of powering a HTPC (Home Theater Personal Computer).

As you might imagine for a program that's been under constant development for the past 13 years, there are a LOT of options for setting up Kodi. There are so many, in fact, that it can be a bit intimidating to get Kodi setup and running as you like. So, let's take a look at getting some of Kodi's system-wide settings configured.

While I'm not a Kodi "expert," I have managed to install and configure Kodi to meet my needs and expectations for a HTPC. I can at least share with you what I've been able to discover thus far. There may even be some of you reading this who know more about Kodi than I've discovered, but I suppose that's true of just about every article I write.

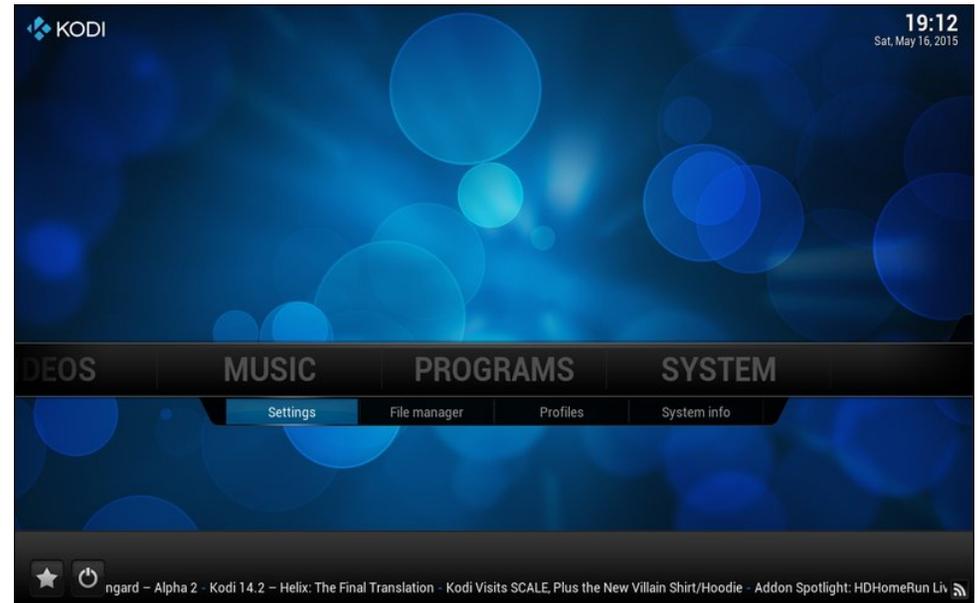
## Install Kodi

Before you can do anything, you will need to install Kodi from the PCLinuxOS repository, and then start it from your PCLinuxOS menu (under the Video menu). When you do, you should see something like the image below. Note that this is an image from the first run of Kodi, and the time really wasn't 11:59 PM. I saved the changing of the time for later in this article.



Kodi uses what's known as a "10 Foot User Interface." This means that the visual interface is intended to be used from a distance of 10 feet, or 3 meters. Next, you're going to want to navigate over to the "System" settings. You can use the mouse, but it can be difficult to stop on your chosen category, especially if your selection is in the middle of the main level categories. Things fly by rather quickly, and I've yet to find a setting to slow things down. You may find it easier (and less frustrating) to simply use the left and right cursor keys on your keyboard to navigate to the category you want to use, and the down cursor key to access the selected category's submenus. If you have a mouse with a scroll wheel, you can also use the scroll wheel to select the main level categories.

Once you've navigated to the "System" category, select the "Settings" submenu. In many cases, the default values will work just fine. Still, it doesn't hurt to know where things are, just in case you want to tweak and personalize Kodi to your own tastes.



## Kodi's Appearance Options

Select the first category, "Appearance."



Notice that at the bottom of the settings window is a control called “Settings level.” Your settings level will be “Settings level: Basic.” I recommend changing this to “Expert” mode, simply by clicking on it. This will unveil several “hidden” settings in the various settings windows you’ll visit.

Under the “Skin” subcategory, you can change the “skin” that Kodi uses. The default is called “Confluence,” and I’ve chosen to keep it. For me, it works and is attractive enough. The rest of the settings should be fairly self explanatory.



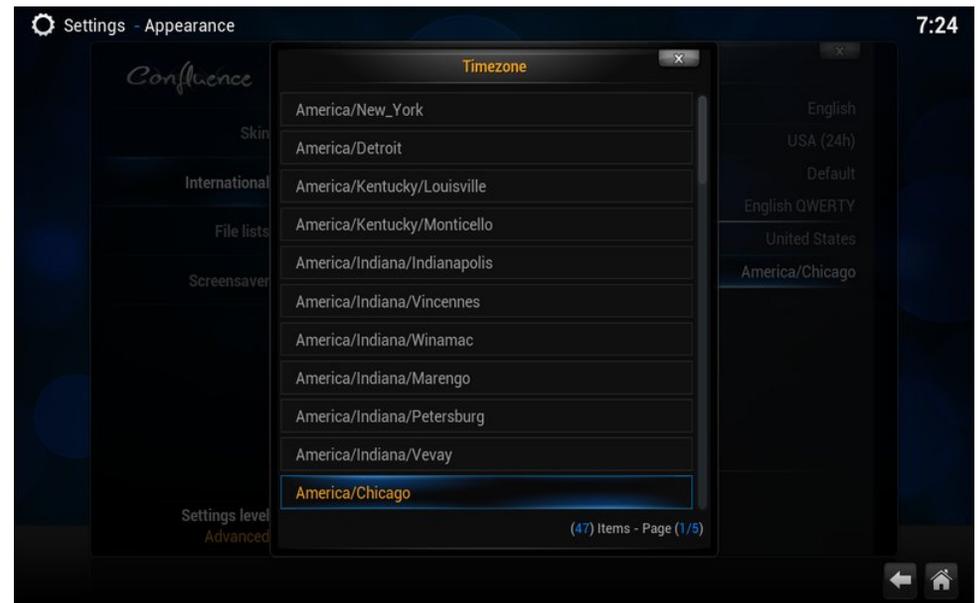
Under the “International” subcategory, you can change the language that Kodi uses. Click on the “Region” setting to set your region (I prefer a 24 hour time format, so I’ve chosen USA (24h) as my setting). The “Character set” and “Keyboard layouts” settings should be fairly self explanatory. Also, if you royally screw up things, you can always click on the “Reset above settings to default” selection to reset everything to the default settings, and start over. You will see this selection appear in many of the Kodi subcategories.



Click on the “Timezone country” setting to select your country from the list.



Then, click on the “Timezone” setting and select the appropriate time zone for your location. If your city or town isn’t listed, just select another city/town from the list that is in your same time zone. Kodi will use the time from your PCLinuxOS installation, but most likely won’t have the proper time zone preselected.



Under the “File lists” subcategory, you can select how Kodi displays files on your system.



Kodi even has its own built in screensaver, and you can control its behavior from the “Screensaver” subcategory. Again, the settings should be rather self explanatory.



## Kodi's Video Options

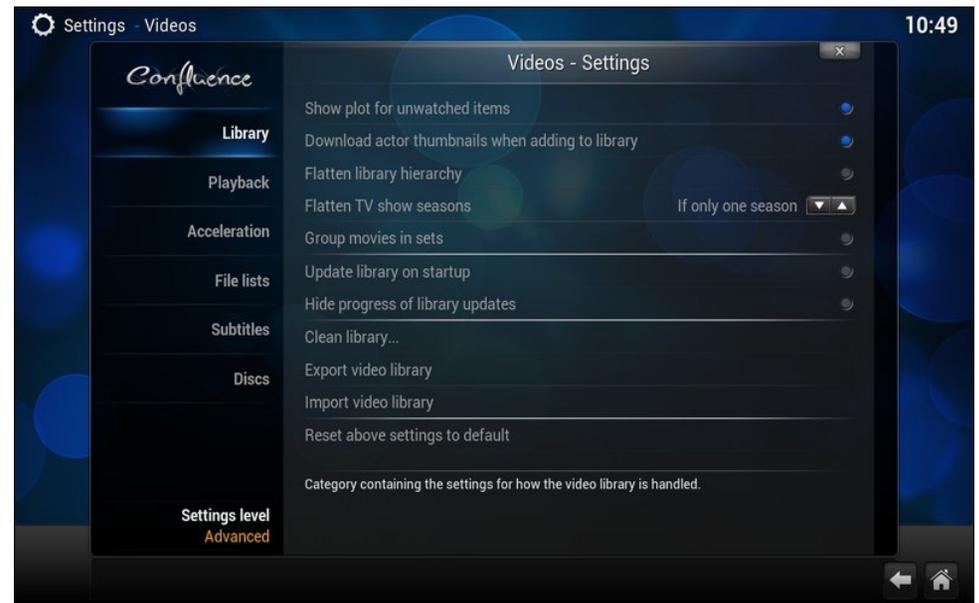
Under the "Video" category, you will find all sorts of settings for telling Kodi how to manage and display video.



The "Library" subcategory refers to videos that you have stored on your computer, whether they are archived from DVDs or downloaded. The image above shows the default values. One item you may want to activate is the "Update library on startup" setting, especially if you are routinely adding videos to your library.

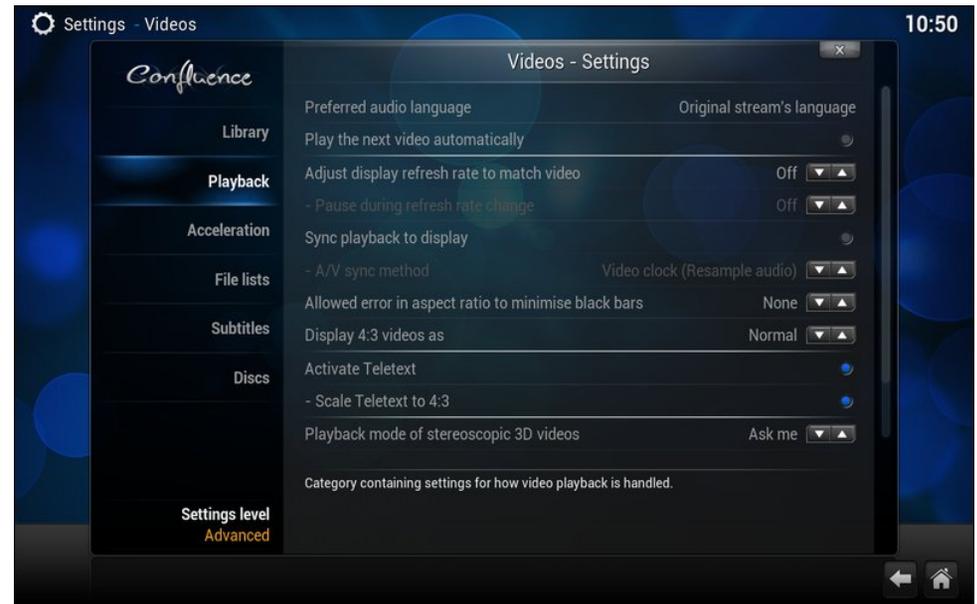
The "Clean library..." setting will cause Kodi to scan your library, and remove any entries that are no longer present, either because they've been moved or deleted. The "Export video library" setting exports your video library to an XML list, while "Import video library" imports an XML list of videos.

To add movies to your library, you will need to add a "scraper" plugin. We'll talk about plugins later.

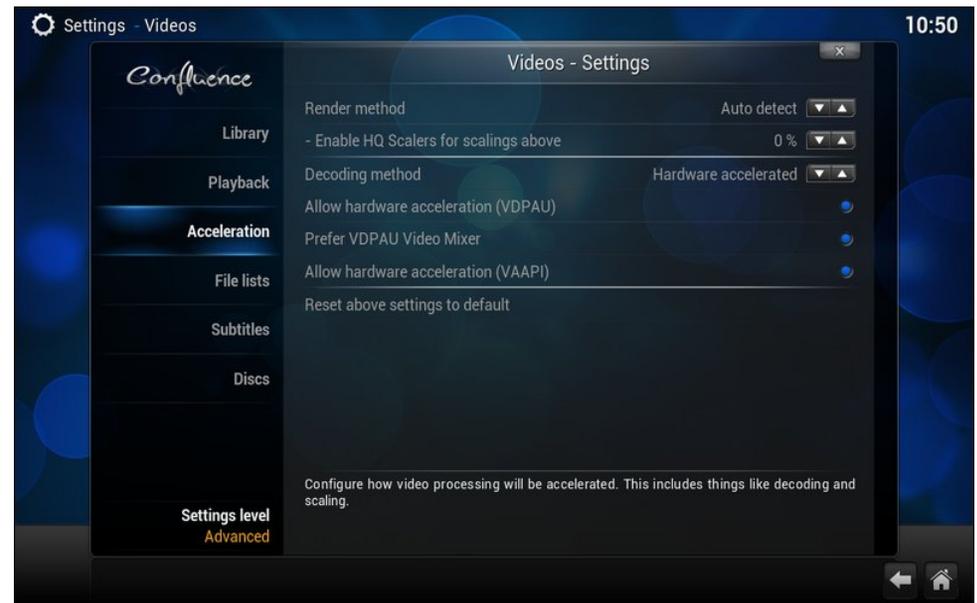


Under the “Playback” subcategory, there are most likely only two settings you will want to change. First, the “Preferred audio language” defaults to “Original stream’s language.” This means if you watch a Bollywood movie, the language will be Hindi. But, if you change it to “User interface language,” it will playback the audio in your current language – but only if that audio stream exists.

The second item you may want to change is the “Display 4:3 videos as” setting. The default value is “Normal.” If you are playing back videos with a 4:3 aspect ratio on your widescreen HDTV, you may want to choose one of the other settings ... or live with the black bars on the left and right side of your screen. Those are Wide Zoom, Stretch 16:9, and Zoom, if you choose to not live with the black bars on each side of your image.



Unless you have a really good understanding of these settings, you might just want to leave them set at their default settings (pictured at right). These settings allow you to adjust the video acceleration method that Kodi uses.

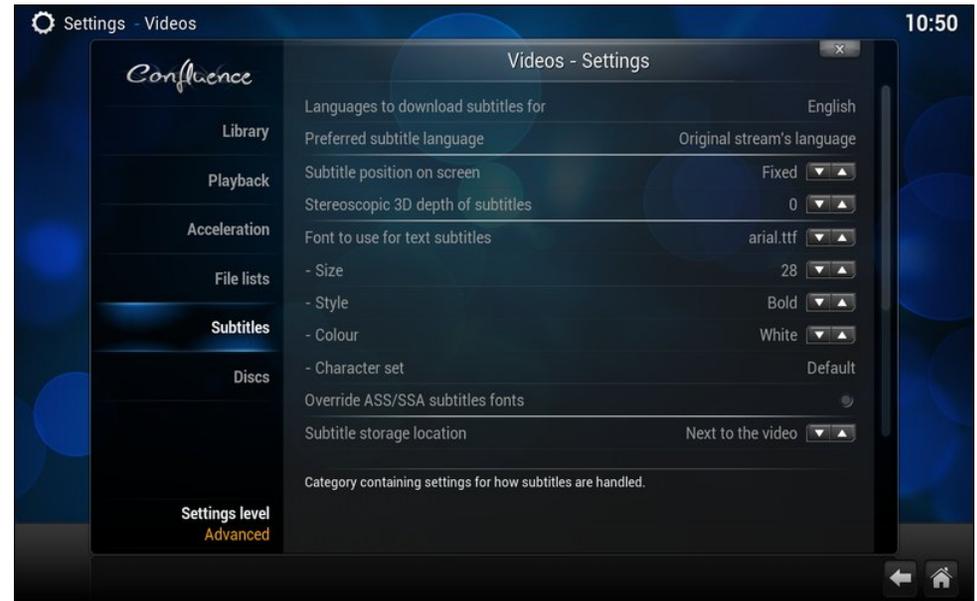
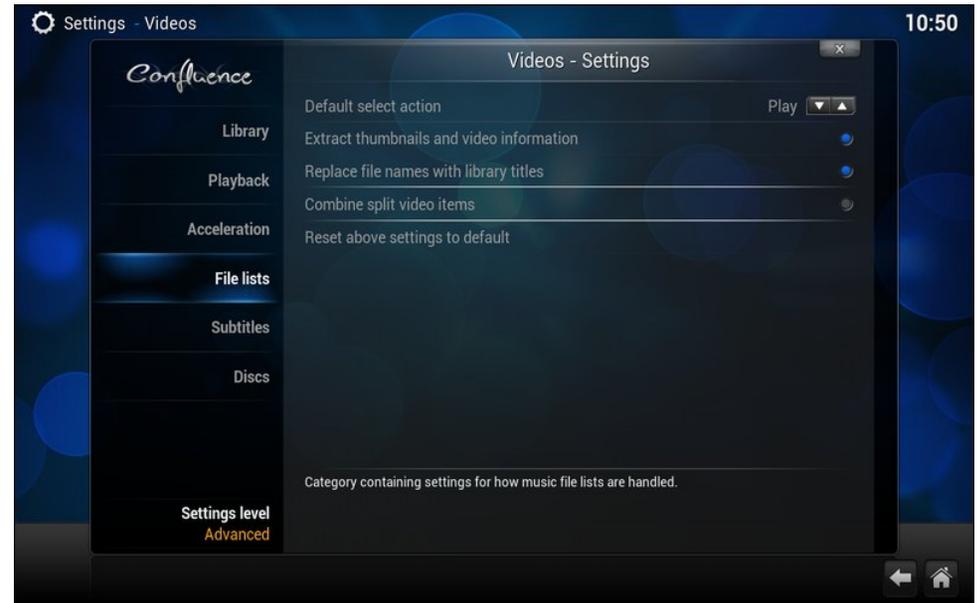


The “File Lists” subcategory allows you to control how Kodi handles files from a file list. The fourth item down – “Combine split video items” – is disabled by default (as depicted above). If you have video file in multipart files, you might want to enable this setting. It will then automatically combine those files and play them as one continuous file. BUT (there always has to be a but, right?), you have to make sure your multipart files are numbered in a certain way.

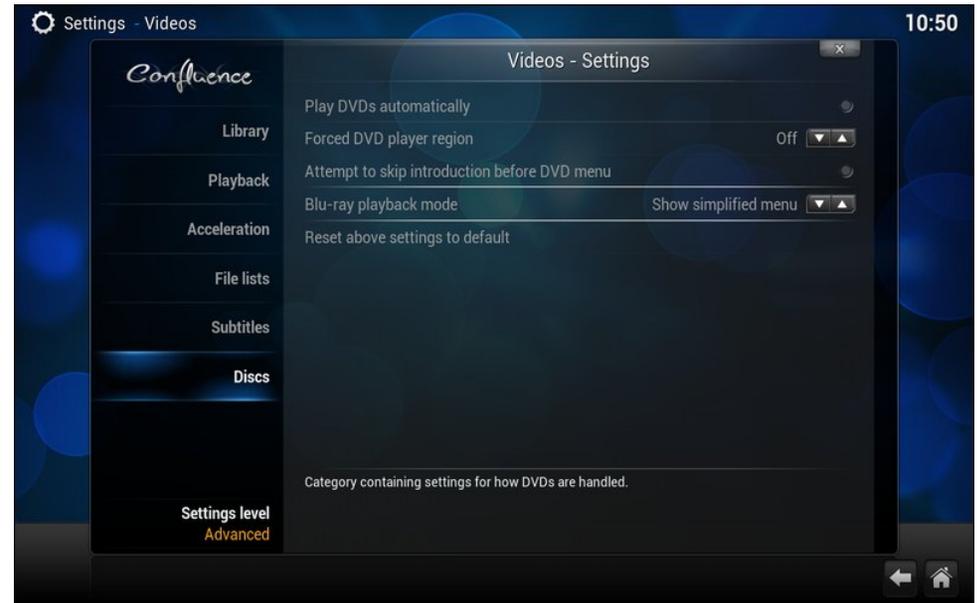
If they are individual video files, you have to insure that they are named and numbered with either part#, cd#, dvd#, pt#, disc# or disk#, where there is no space before the #, and where # is any character between 1 through 9, and/or A through D – and they have to be in the same directory. You can also use moviename#.ext, where number may be any character A through D. Thus, if you had Star Wars stored as a multipart file, using the naming convention StarWarsPart1.mp4 and StarWarsPart2.mp4 would enable Kodi to play both parts (starting with Part1) as one continuous file. Similarly, StarWarsA.mp4 and StarWarsB.mp4 would accomplish the same thing.

Kodi can also do something similar with video files in folders, such as VIDEO\_TS folders representing DVD rips that are not in standard ISO format. You can stack the folders using a name that contains the movie name, ending with cd#, dvd#, disc# or disk#, where there is no space before the #, and where # represents a number between 1 through 9. For example, if you have folders named The Lord of the Rings - The Fellowship of the Ring DVD1/VIDEO\_TS/etc... and The Lord of the Rings - The Fellowship of the Ring DVD2/VIDEO\_TS/etc... will appear as a single entry in both the library and file views.

The “Subtitles” subcategory does exactly what you would expect it to do – it controls the display of subtitles, should you choose to turn them on or use them. The settings should all be self explanatory.



The final Videos - Settings sub category, "Discs," allows you to control playback of both DVD and Blu-ray video discs. The default settings are as displayed in the image above. You may wish to enable the two settings that are set to off by default. If you insert a DVD disc while Kodi is running, the DVD will begin playing automatically – if you enable the "Play DVDs automatically" setting. If you enable the "Attempt to skip introduction before DVD menu" setting on, Kodi will attempt to skip straight to the DVD menu, so that you don't have to sit through all the BS they put on DVDs these days before you can even watch the disc's content.

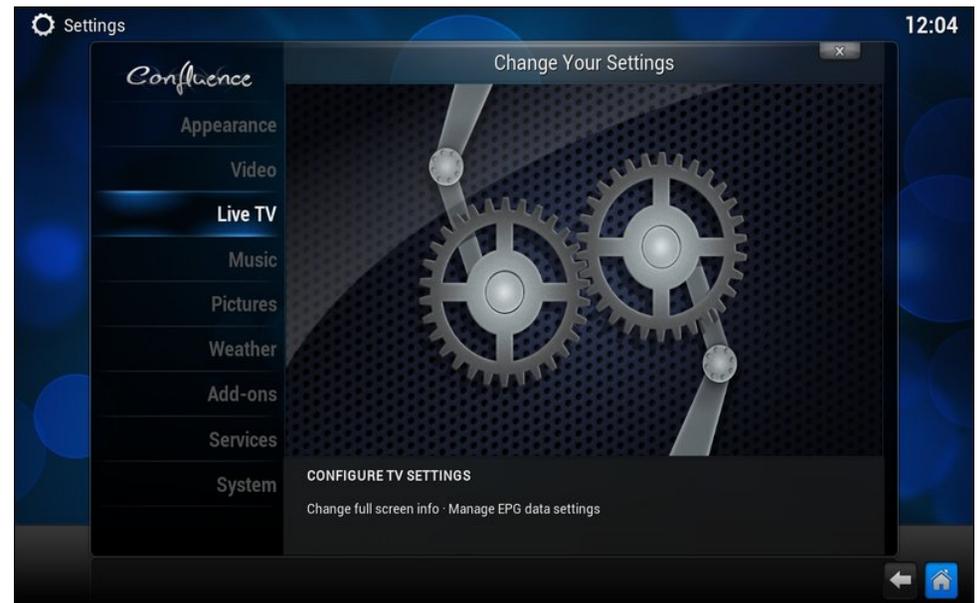


## Kodi's Live TV Options

To be fair, I'm going to defer going over the Live TV settings until a later date. My HTPC currently does not have a tuner, and without one, I don't feel that I can provide an accurate accounting of the settings in this category.

When I built my HTPC, I did (do) have a tuner card (that I bought a couple of years ago, but never used) ... but it is a full height card, and the case of my HTPC only accepts half-height cards. So, I've purchased a USB TV Tuner from Newegg that has gotten excellent reports of working on Linux and with Kodi. Once I get the new TV tuner set up, I hope to be able to revisit the Live TV settings.

Yes, there are IPTV (Internet Protocol TV) stations. But it has been my experience that they are somewhat flaky, and only work sporadically. In fact, I find more that don't work at all than any that do work at all. They are definitely not reliable enough, in my humble opinion, to depend upon as your only source for live TV. With a compatible TV Tuner card (or USB stick, in this case), you should be able to receive OTA (over the air) channels, as well as input from a cable provider's set top box.

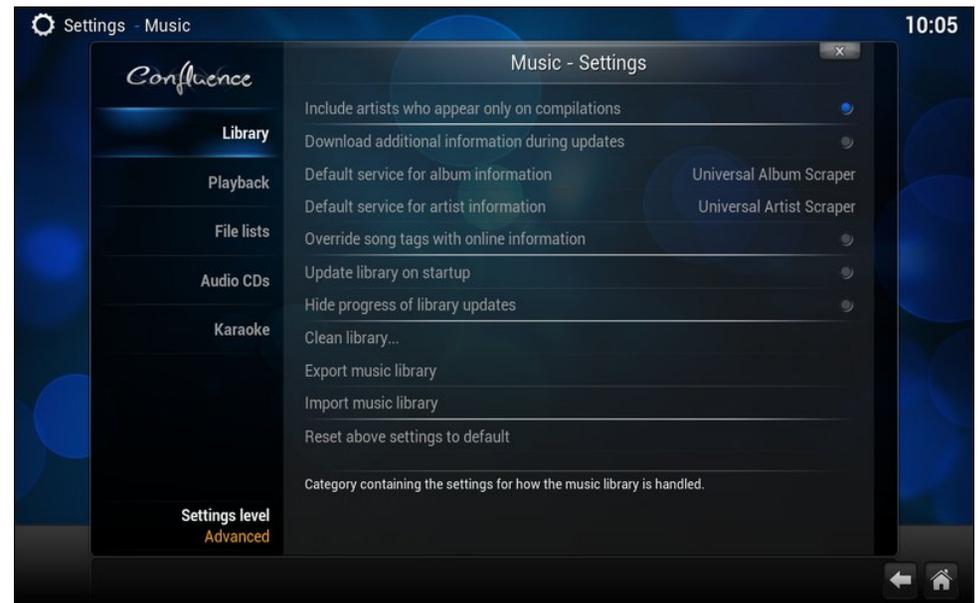


## Kodi's Music Options

Select the "Music" category, where you can tell Kodi how to handle the music files you ask it to play.



The "Library" subcategory for Music is very similar to the Video > Library subcategory ... with one huge exception: the "scraper" plugins are already installed and ready to go. The settings here should be fairly self explanatory, or similar enough to the Video > Library subcategory to avoid repeating here.



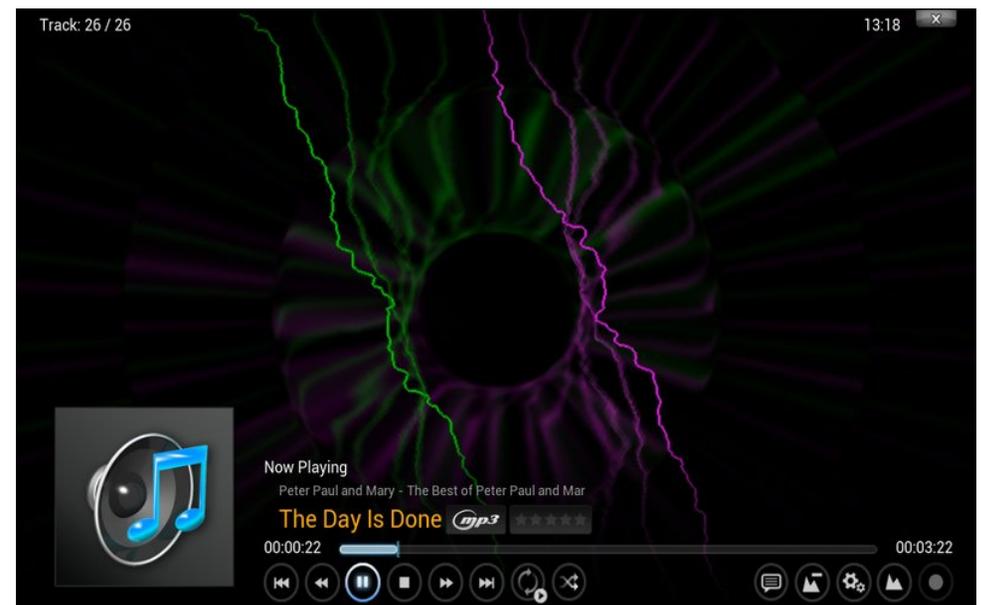
Under the “Playback” subcategory, you can tell Kodi important information about how you want to play your music files. There are a few settings you may want to change. The default values are shown above.

First, you may want to turn on the “Queue songs on selection” option. This will allow your songs to be queued when you click them, rather than played. Using this option, you can customize your playlist more easily.

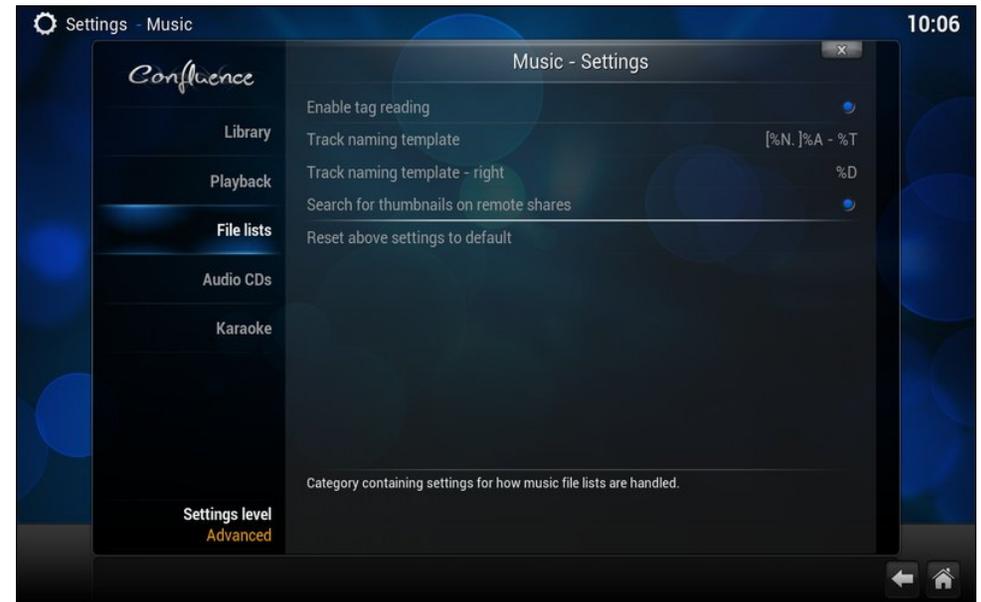
Second, you may want to turn on the “Crossfade between songs” option. You can set the number of seconds for the overlap. In many cases, it creates a very smooth transition between songs. You can also set it to only crossfade between songs on the same album.



Third, you may wish to change the visualization that is displayed when your songs play. I've changed mine to the “FishBMC” visualization (pictured at right). There are several available pre-installed, or you can get and install many others by clicking on the “Get More...” button. Of course, you can turn off all visualizations, if you like, too.

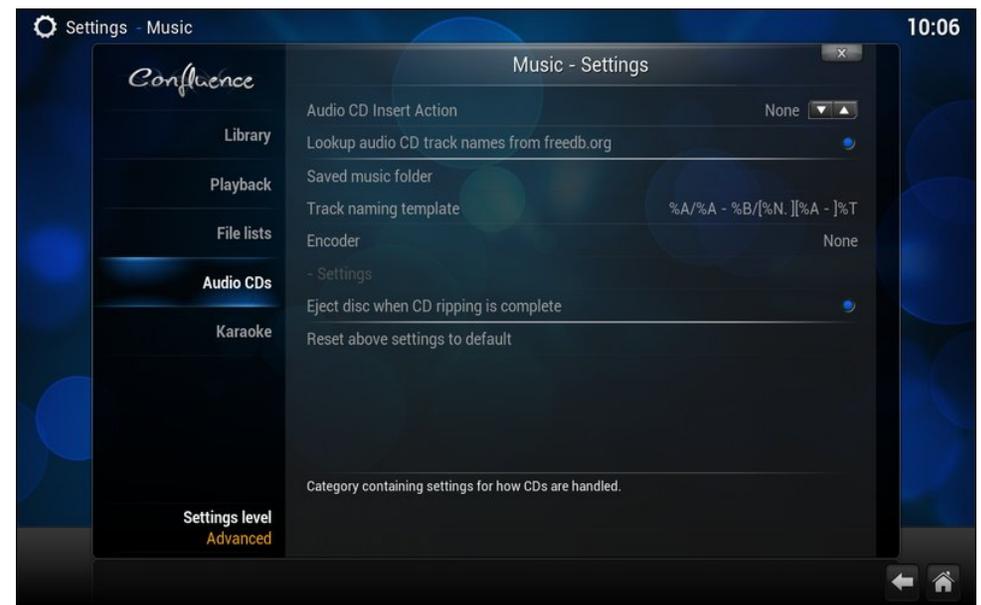


The “File lists” subcategory allows you to set the naming conventions that Kodi uses for your music files. Anyone who has dealt with music files should find the settings rather self explanatory.

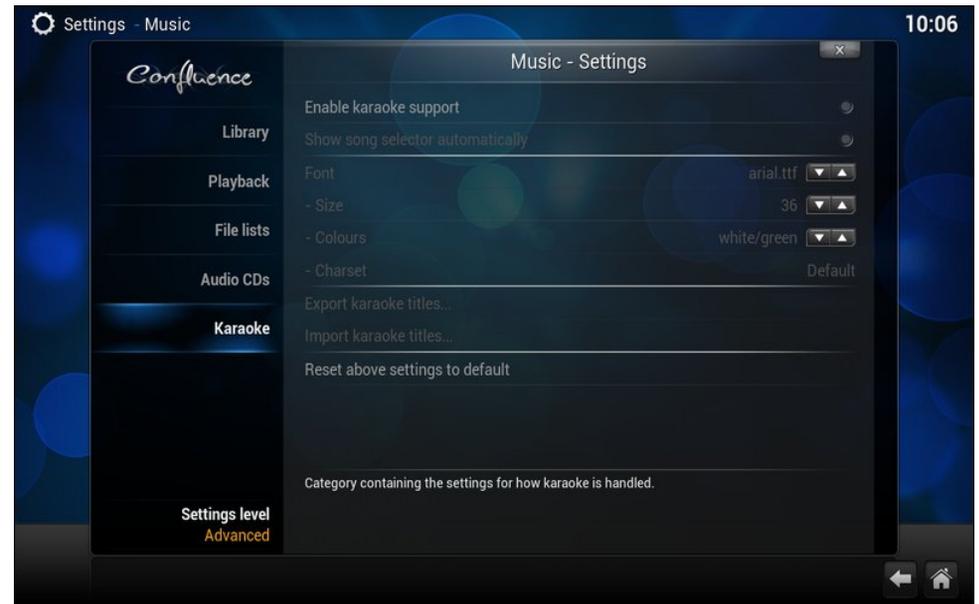


I bet you can't guess what the “Audio CDs” subcategory is for? Yep. That's right. It tells Kodi how to handle audio CDs. The default values are shown in the image above. If you want, and Kodi is running, you can have Kodi do nothing, automatically play the CD, or rip the CD. If you want either of the latter two options, you will need to change the default value of “None” to either “Play” or “Rip,” depending on what you want Kodi to do. Whichever setting you make in Kodi, it will look up your audio CD's track names from freedb.org, by default. Of course, you can turn this feature off, if you wish.

Especially important if you are going to have Kodi set to rip your audio CDs, you will need to tell Kodi where to store the ripped audio files. You will also have to set the naming convention you want Kodi to use for naming the folder(s), files, etc., if you have other ideas than the default values. (Hint: hover over the “Track naming template” setting to see a key that describes what each variable stands for.)



“Karaoke” is the last subcategory under the Music settings screen. I really can’t comment much on the settings here. I simply don’t “do” karaoke. I can’t carry a tune in a lead box. If I tried to sing, dogs would howl in unison across multiple continents. Thus, I’ll leave the discovery of what these settings do for those of you who sing, and who are inclined to explore these settings.

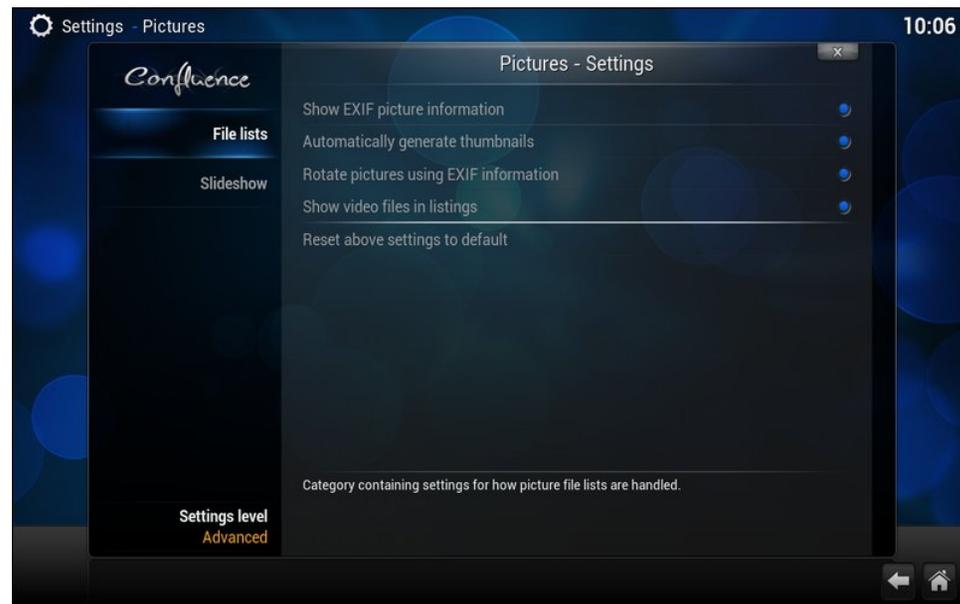


## Kodi's Picture Options

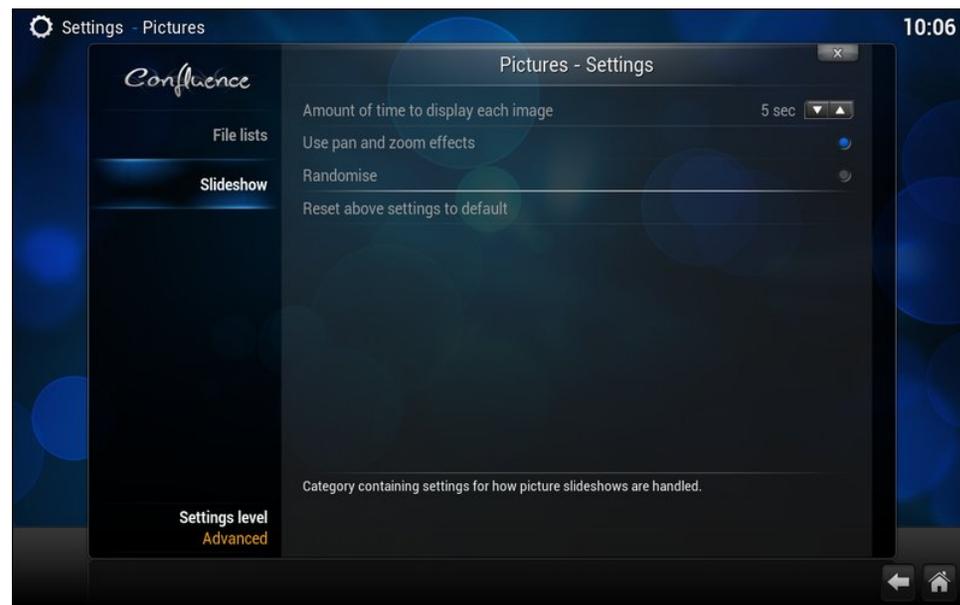
Select the “Pictures” subcategory to set up how Kodi displays pictures.



You'll most likely want to keep the default settings, as pictured on the right. The settings should be self explanatory.



Kodi will even display your images in a slideshow. Again, the settings here should be rather self explanatory.



## Kodi's Weather Options

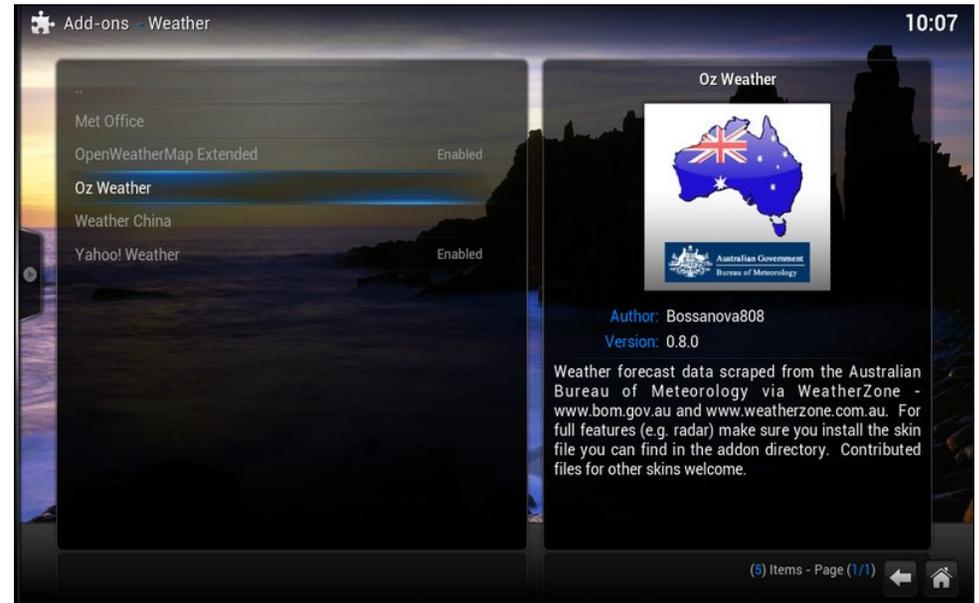
Here, you can setup Kodi to display a weather forecast for your area.



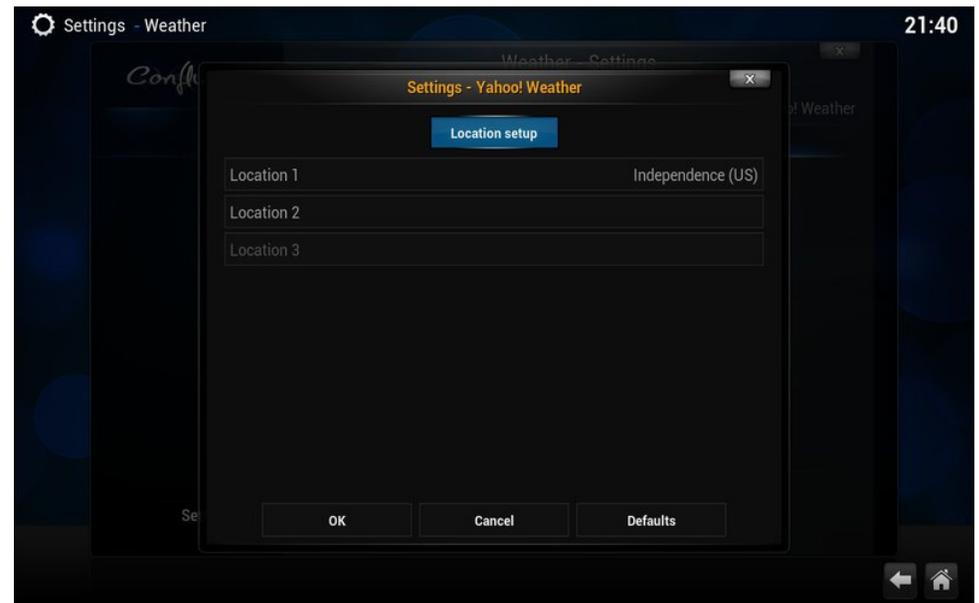
Clicking on the "Service for weather information" setting will display the screen depicted at right. The Yahoo! Weather service has more cities to choose from, but the OpenWeatherMap Extended service will display various weather maps. If you choose the latter, you will have to pick a "large" city nearest to your location.



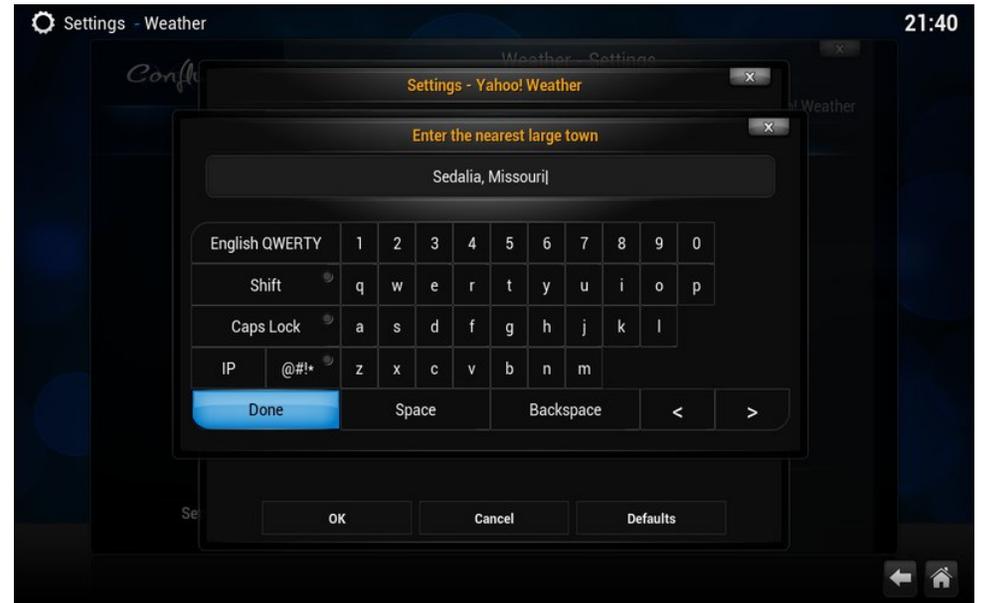
There are three other services to choose from ... if you click on the "Get More..." button in the previous screen.



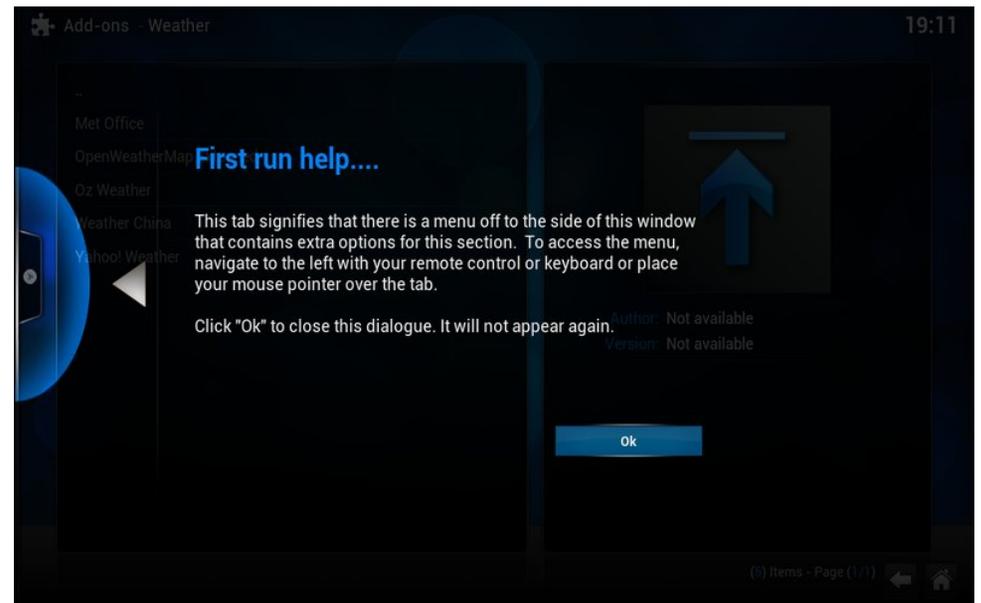
To set the location, simply click on the location you want to configure.



Then, enter the town that you want to add in the top text entry box. When finished, click on the “Done” button.



The first time you see the pop out menu on the left, you'll get a one time instruction screen (pictured at right). Its contents will vary, depending on which main level menu entry you're currently using.



This is what the Yahoo! Weather service looks like when you select “Weather” from the main level menu.



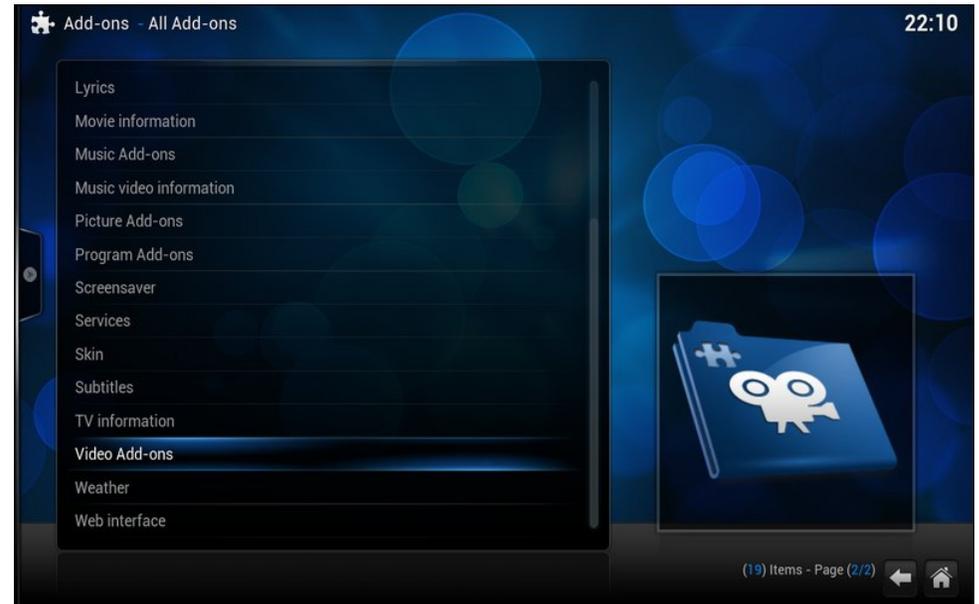
## Kodi's Add-ons Options

Here, you can setup Kodi with what seems like an endless number of add-ons.



It seems as if Kodi runs entirely on add-ons and plugins. Once you choose the Add-ons category, you'll see a screen similar to the one above. Each item represents an entry to a world of more choices, which can often lead to even more choices. Don't fret if you begin to feel lost as you go deeper and deeper into each category. The back arrow in the lower right corner of the screen will take you back one level, and selecting it multiple times will take you back multiple levels ... all the way to the main level menus, if you keep going.

We'll cover specific add-ons as we go through each main level menu of Kodi. But don't be afraid to explore. For example, under the Screensaver add-on subcategory, you can choose from several additional screensavers. Just click on the add-on you want to install, and Kodi will download and enable the chosen add-on. There's just no way we (I) can go through every add-on that's available.



## Kodi's Services Options

I doubt that the majority of users will find much need to change many of the settings in this category of settings. I know I didn't. The things that are configurable here include the device name (default is Kodi), setting up UPnP services (on by default), setting up a webserver that allows for control of Kodi via HTTP (off by default), setting up "Remote Control" (defaults are "On" for programs on the installed system, and "Off" for programs on other systems), setting up Zeroconf (default is on), allowing Kodi to receive AirPlay content (off by default), and setting Kodi up to connect to a SMB (Samba) server.

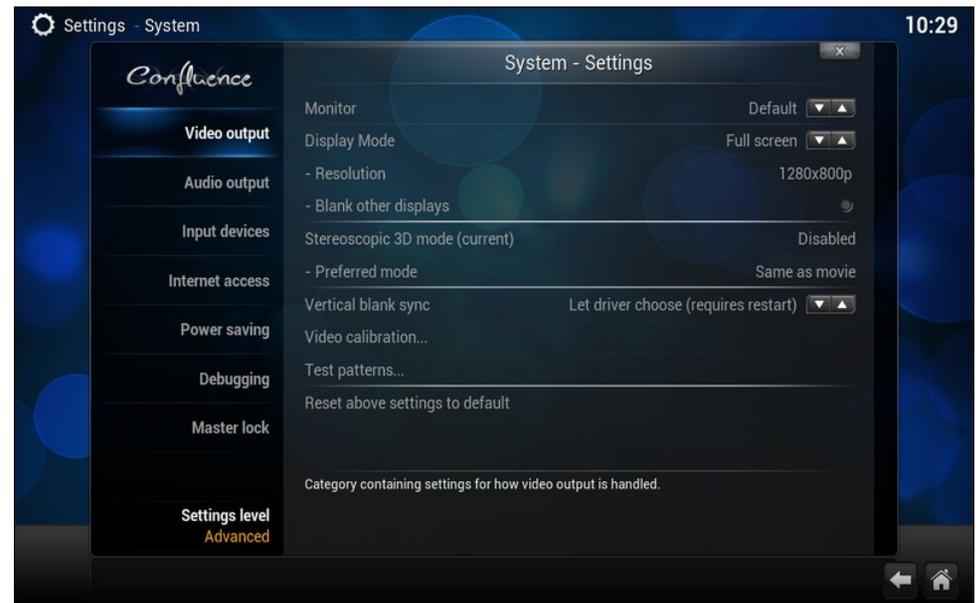


## Kodi's System Options

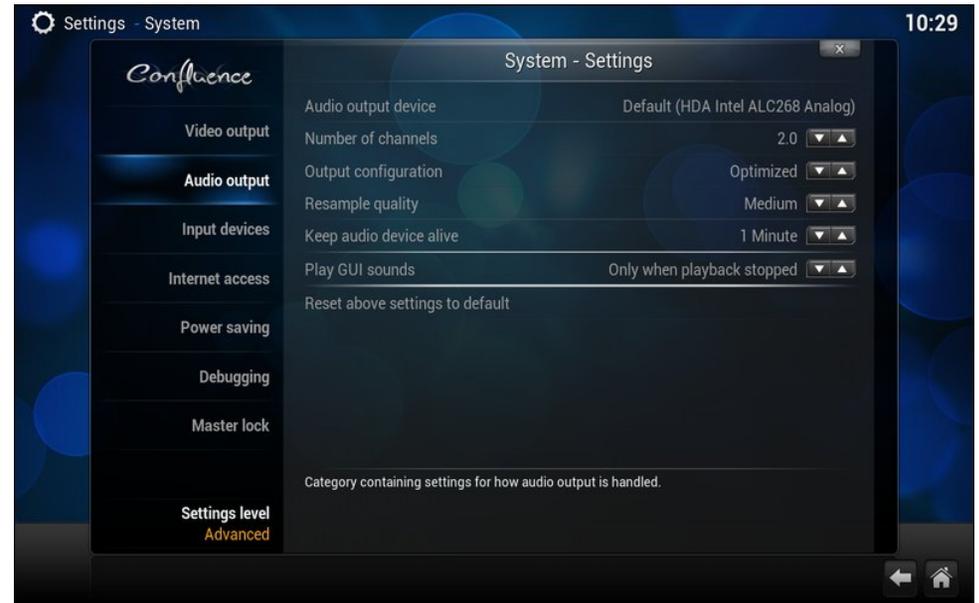
Under the "System" category, you can setup and calibrate your displays, configure your audio output, setup remote controls, set power savings options, enable debugging, and setting up a master lock for your Kodi installation.



Under the "Video output" subcategory, you can determine how Kodi handles video output. Typically, you should be just fine accepting the defaults in this section, as Kodi will use the video configuration of the base system (which is PCLinuxOS for our purposes). However, if you go to setup Kodi using something like OpenElec as the base system, you might have a need to tweak these settings a bit.



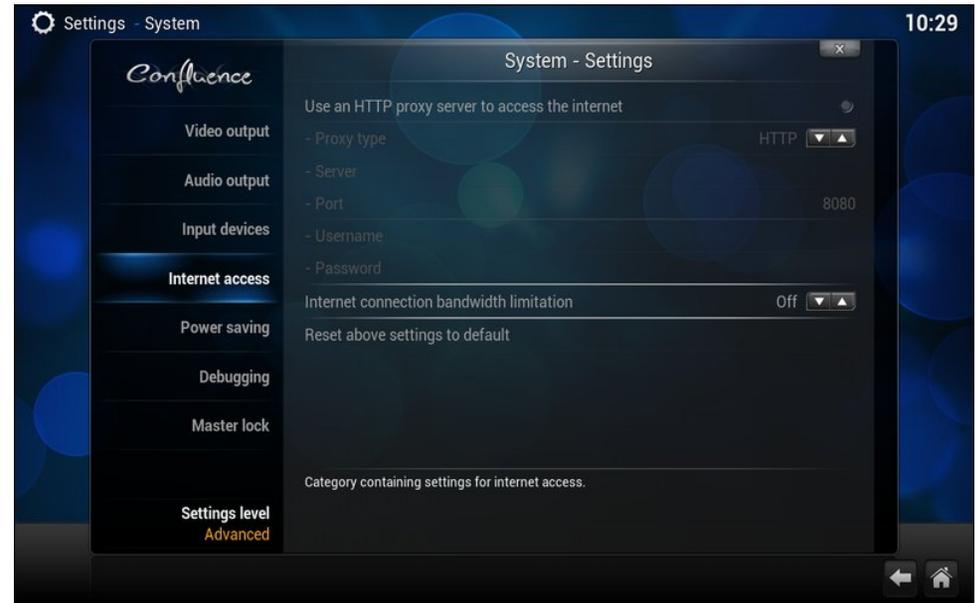
Under the “Audio output” subcategory, you can setup Kodi for how audio output is handled. Just as with the previous subcategory, Kodi will inherit the audio output settings from your base installation of PCLinuxOS. The extra settings in this section should be fairly self explanatory.



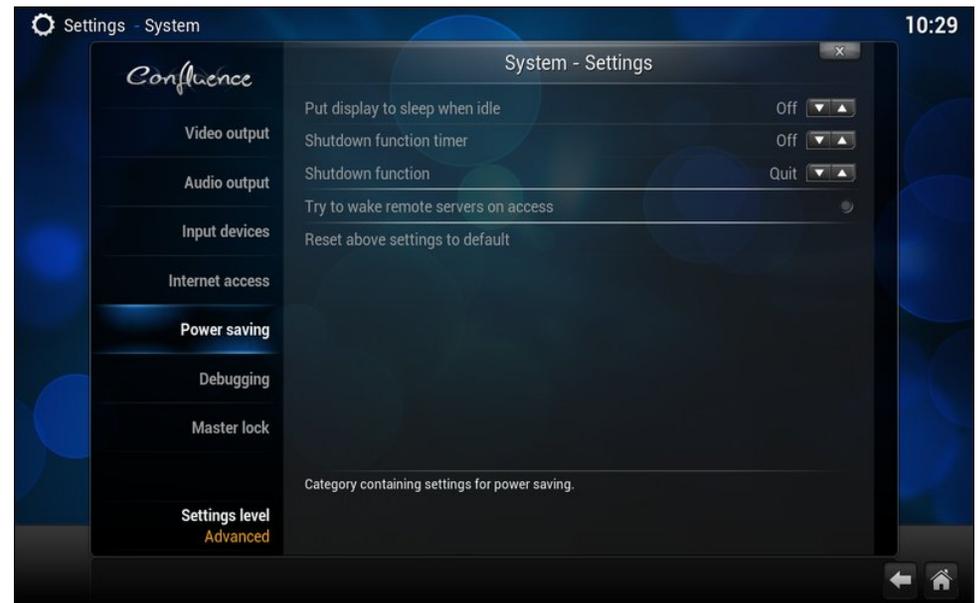
Under the “Input devices” subcategory, you can configure how Kodi handles input devices. Both of the available settings are on by default, and I’ll forgo any further explanation of these settings, since they should be very self explanatory.



Most people probably won't have to change anything under the "Internet access" subcategory – unless you use a HTTP proxy server to access the internet. You can also limit how much of your internet connection bandwidth Kodi is allowed to consume. This latter setting may be important if you are using a shared internet connection and you find that Kodi is consuming the lion's share of your internet connection's bandwidth – and leaving very little for any other computers sharing that connection. The default settings are shown in the image above.

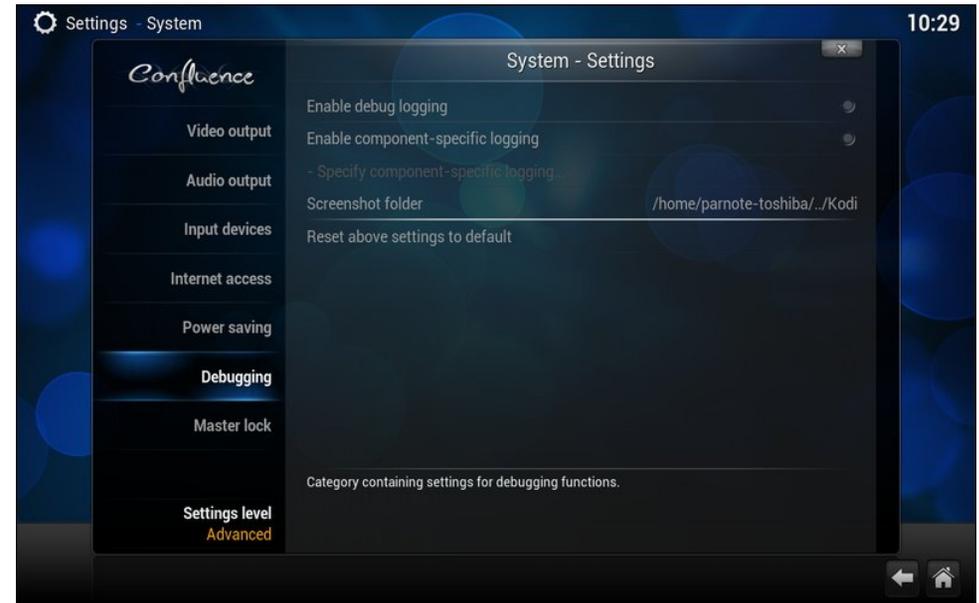


Under the "Power Saving" subcategory, you can setup how Kodi handles power saving settings. The default values are shown in the image above, and the settings should be self explanatory.

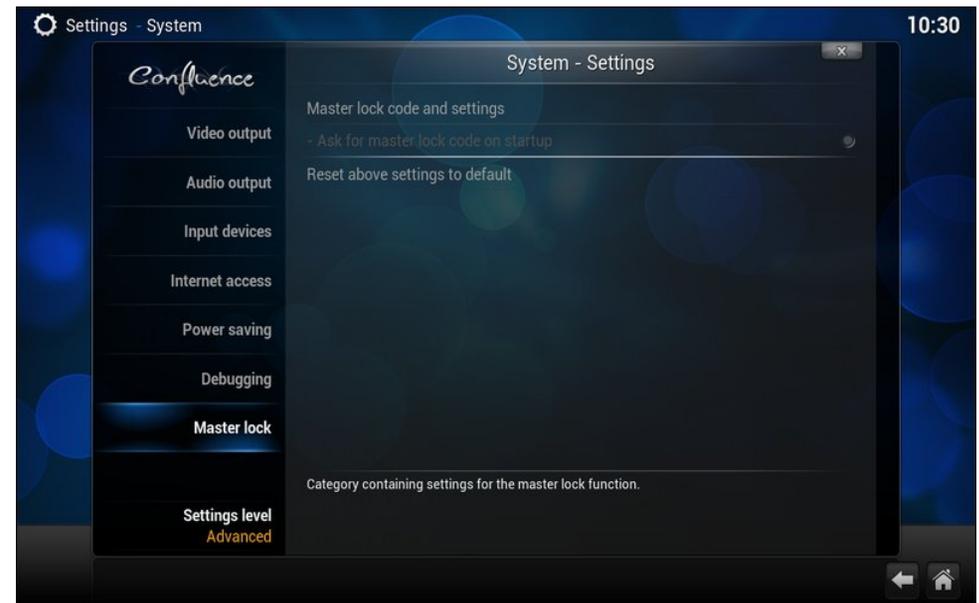


While I doubt that there are many users who will need to change the settings in the “Debugging” subcategory, I can easily see a few instances when you might. If, for example, you were having some issues with Kodi and you were seeking help with those issues on a Kodi user forum, someone may ask you for the debugging information to help track down the cause of your issue. So, you will need to go into this section to turn on the debugging settings. The default values are shown in the image above.

One other possible reason to access this section would be if you want to change the folder where you have Kodi saving screenshots. To me, it seems like placing the setting for the screenshot folder here is an afterthought. I know I would NEVER think to look here for that setting. Plus, what the screenshot folder designation has to do with debugging is far beyond my comprehension. It would seem much more intuitive to place this setting under the Video, Live TV or Pictures category – and NOT the System > Debugging section. I guess this is just one of those things that make you go “Huh?” and defies logical explanation.



The final subcategory is “Master lock.” The default values are as shown in the image above. While the settings in this section should be self explanatory, why you might want to use this may not be so evident. If access to your HTPC is unfettered – meaning, it’s accessible to all members of your family or household – you may not want just anyone making changes to how Kodi is set up. If this sounds like something you want to avoid, you may want to “lock things down” so your kids/spouse/roommates/etc. can’t install and uninstall addons and plugins, or otherwise change the functionality of Kodi.



**Summary**

Like I mentioned at the start, there are a LOT of settings for Kodi. This stands to reason, for a 13 year old program that does so much in the area of multimedia. Believe it or not, we've just started to scratch the surface. Each individual main level category has to be set up independently, as well.

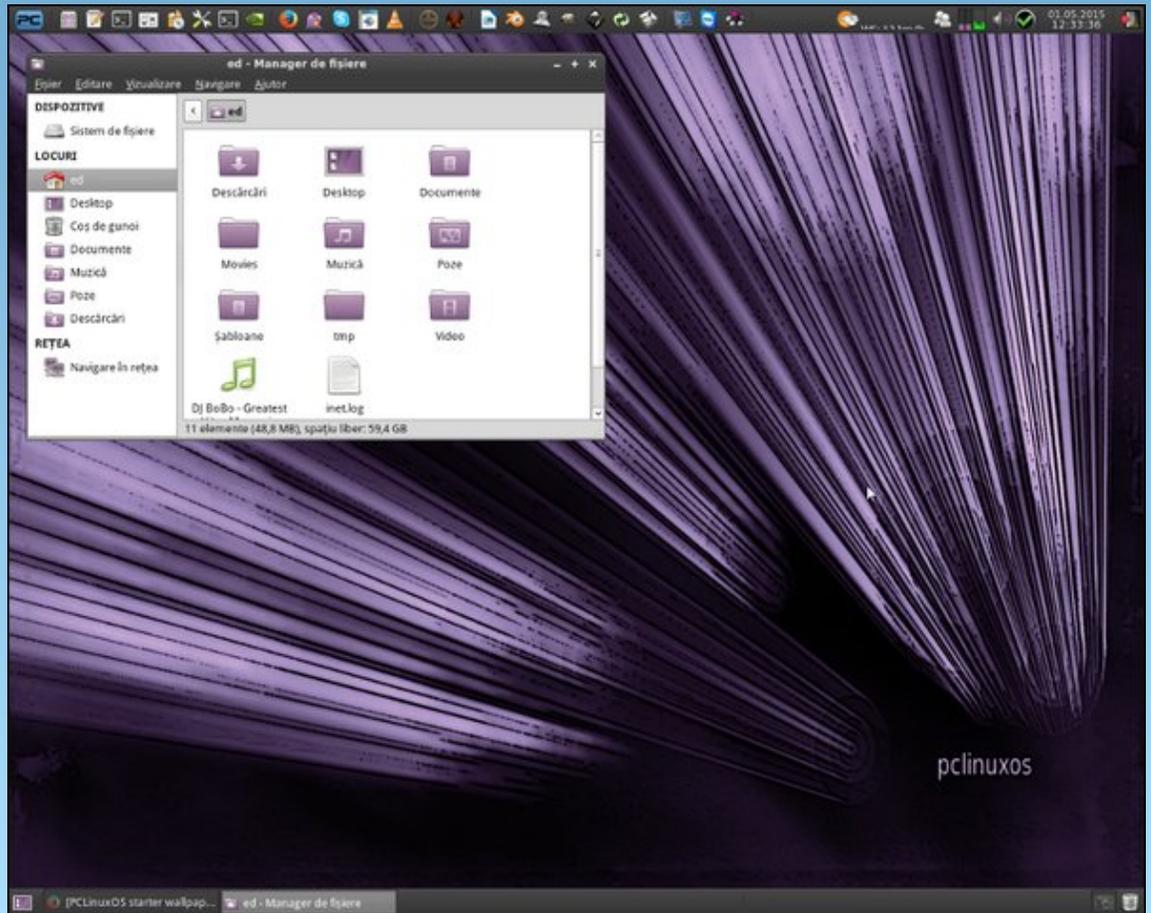
We'll take a look at how to set up each individual section of Kodi in the coming articles. Because of what all is involved in setting up each section, we'll focus each article on setting up one or two sections at a time.

Yes, setting up Kodi is a lot of work. But, in the end, it's worth it. For the most part, with the exception of routine maintenance, this is something that you will have to do once. When it's all said and done, you will be enjoying many hours of entertainment bliss with Kodi.

Stay tuned!



# Screenshot Showcase



*Posted by Orion, May 1, 2015, running Xfce.*



A magazine just isn't a magazine without articles to fill the pages.

If you have article ideas, or if you would like to contribute articles to the PCLinuxOS Magazine, send an email to:  
[pclinuxos.mag@gmail.com](mailto:pclinuxos.mag@gmail.com)

We are interested in general articles about Linux, and (of course), articles specific to PCLinuxOS.



# PCLinuxOS Recipe Corner



*from the kitchen of  
youcantoo*

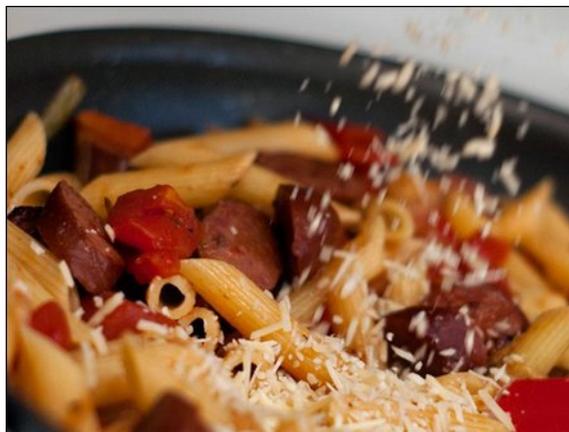
## **Penne with Smoked Sausage and Caramelized Onions**

### **Ingredients**

- 12 oz. smoked sausage, sliced
- 8 oz dry penne pasta
- 1, 14.5 oz can diced tomatoes
- 1 medium onion, sliced
- 1/4 tsp salt
- 1/4 C parmesan cheese, shredded
- 1/2 tsp dried oregano

### **Instructions**

1. In a large skillet brown sliced sausage until caramelized. Remove from skillet and set aside, leaving the drippings from the sausage in the pan.
2. Lower heat to medium-low and place onions and salt in sausage drippings. Cook onion, stirring often, until very soft and and well caramelized-- about 15 minutes.
3. Boil pasta according to package directions and drain, reserving 1/2 C of the starchy cooking water.
4. Add sausage to the onions as well as the tomatoes and oregano. Bring to a simmer and stir in pasta and parmesan cheese, adding a bit of the cooking water if necessary to loosen it up. Serve with additional parmesan cheese and chopped fresh basil if desired.

A collection of PCLinuxOS merchandise. On the left side, there is a white polo shirt, a white mug with the PCLinuxOS logo and the tagline 'radically simple', a circular logo with the PCLinuxOS logo and 'radically simple', and a square logo with the same text. On the right side, there is a black t-shirt with the PCLinuxOS logo, a tan baseball cap with the PCLinuxOS logo, white shorts with the PCLinuxOS logo, and a white t-shirt with the PCLinuxOS logo.

**Support PCLinuxOS! Get Your Official  
PCLinuxOS  
Merchandise Today!**

# *ms\_meme's Nook: PCLOS Forever*

You may think I'm a bit foolish  
You've heard that I'm wild and insane  
You may wonder how I can sing this song  
A gal that has lost half her brain  
But if you just listen to me  
And you know I never would lie  
As sure as I live a song I will give  
About PCLOS until the day that I die

I'm gonna love it forever  
Forever and always amen  
As long as Texstar purrs like an old tom cat  
And in the sandbox they post about this and that  
If you wonder how long I'll be faithful  
I'll be happy to tell you again  
I'm gonna love it forever and ever  
Forever and ever amen



They say time takes its toll on the users  
To another OS they may roam  
Well honey I don't care what the others may do  
PCLOS will always be my home  
They say that time can change the situation  
Nobody knows what the future may bring  
Well it's easy to see it's not a happenin' to me  
About PCLOS I always will sing

I'm gonna love it forever  
Forever and always amen  
As long as Texstar purrs like an old tom cat  
And in the sandbox they post about this and that  
If you wonder how long I'll be faithful  
Just listen to how this song ends  
I'm gonna love it forever and ever  
Forever and ever amen

**MP3**

**OGG**



# Monitor Your System With Conky, Part One

by Peter Kelly (critter)



“Conky is a free, light-weight system monitor for X, that displays any information on your desktop.” That description is from the official website and rather understates the abilities of this useful little application. The name, the website informs us, “comes from a character in the Canadian TV show titled “Trailer Park Boys,” a ventriloquist’s dummy I am told. Don’t let that put you off though. Conky is a serious and effective utility. The idea was forked from a similar tool named “TORSMO”, an abbreviation of “Tyopoyta ORvelo System Monitor” which means something in Finnish and I am sure that author thought it appropriate. Conky is available from the PCLinuxOS repositories.

Today, Conky is a mature program offering a great number of configuration options and access to many system features, such as memory usage and network throughput as standard. Additionally Conky can execute shell functions and output the results on screen. Provision is made for conditional output in form of if-then-else type statements, and the newest version in the repositories now has hooks for the Lua scripting language. This allows the more intrepid users to take advantage of some advanced features, which include the use the Cairo graphics libraries. One feature that seems very popular with users is to have Conky display its output with a transparent background seemingly pasted onto the desktop wallpaper.

The popularity of Conky is such that there are a plethora of examples and downloadable scripts available on the internet, enabling less ambitious users the ability to enjoy some of these extended features.

Conky is not limited to displaying boring system statistics. You can display a clock, graphics, weather reports, or whatever your imagination can dream up. You can see some advanced examples at <http://www.deviantart.com/browse/all/?q=conky>. Many of the examples on this website have accompanying code that can be downloaded to use on your own machine. Many of the examples use some very advanced techniques not available with the basic Conky which will not be discussed until the second part of the article, by which time I hope to have covered enough theory to allow you to compete with them.

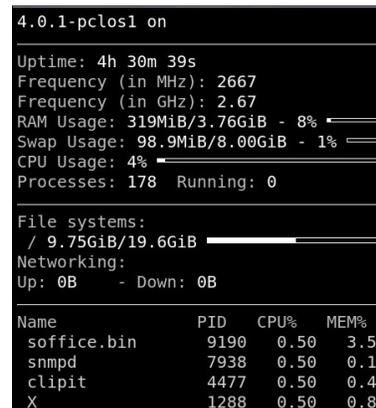
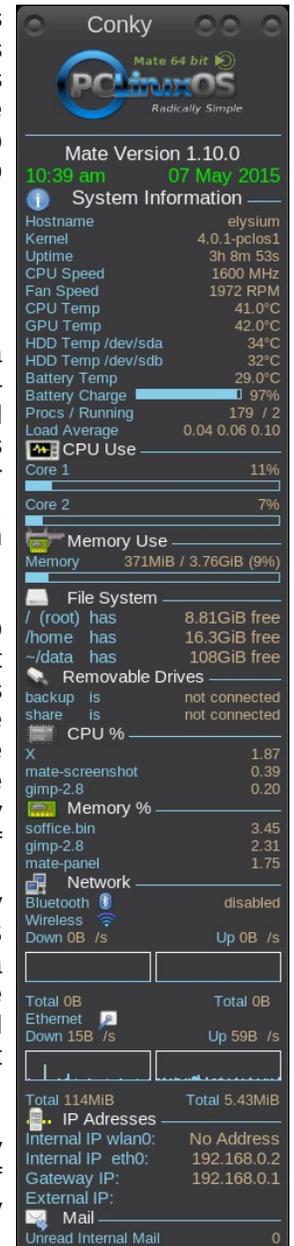
This article will start by showing you some less demanding conky examples, take you through the basics of setting up your own Conky displays, as well as demonstrate the use of shell commands to generate output. Part two will introduce the use of interfacing to Lua scripts and exploiting the functions in the Cairo libraries.

## Getting Started

To get started I’ll show, step by step, how to build a useful system monitor based on one I use myself – shown at right but without transparency. The original format for this file I picked up from somewhere on this forum a while back, but unfortunately I don’t remember who posted it. Whoever it was, thank you. The content, however, has been extensively reworked to suit my own requirements.

A Conky configuration file is a plain text file in two sections, both of which follow a strict format. The first section is the configuration section which determines how and where the Conky window and its contents are displayed. The second section begins after the word ‘TEXT’ on a line of its own. Everything after this line is a candidate for display. I say candidate as it will only be displayed correctly if the format is correct. After installing Conky from the PCLinuxOS repository open a terminal and type the word **conky**. You should see the window at left displayed.

Some of the figures may be moving about, but if you see this, then Conky



is working. To generate this screen, Conky uses its default configuration file which is:

```
/etc/conky/conky.conf
```

You could copy this file to your home directory:

```
cp /etc/conky/conky.conf ~/.conkyrc
```

and modify it, or you can create your own from scratch, which is what I am going to do.

When the conky command is executed, it looks first for a hidden file in your home directory called `.conky` and, if found, uses that. If it is not found, then it uses its default file. That is, unless you start Conky with the `-c` option. This option tells Conky to use a named configuration file, which must be the next thing along on the command line before you press the enter key. This allows you to have multiple configuration files and multiple instances of Conky running, all doing different things. I'm going to name my configuration file `.conkyrc_mag`, so the start up command then becomes:

```
conky -c ~/.conkyrc_mag
```

I got into the habit of using the leading `~/` in case I type the command when I am not in my home directory and this is also the way you should use the command in an auto-start routine.

### The Configuration File – Section 1

The commands to be used in the first section are all listed in the official documentation, which can be found at [http://conky.sourceforge.net/config\\_settings.html](http://conky.sourceforge.net/config_settings.html). However, the commands and the accompanying explanation can be confusing, unless you are a seasoned Conky script writer. Some of them we'll never use anyway. This is how I configured things for the image above.

It perhaps seems to be a lot to set up, but most of it will be the same in all of your configuration files with only minor changes. So once you have a set up that you like, there will need to be only minor changes, and some of the lines can be safely left out. Many of these options can also be set on the command line. Type **conky --help** for a full list.

Some of these options concern transparency that works only when you are using a window manager that supports compositing and have that feature enabled. Fortunately, most of the window managers used by PCLinuxOS releases have

this feature, but it **must** be enabled for transparency to work! In Mate, which I am currently using, this is done from the Mate Control Centre > Windows module where there is a checkbox to enable or disable the feature. Under KDE, you must enable desktop effects.

Here is the configuration section of `~/.conkyrc_mag`. The numbers on the lines below are not part of the file and should not be entered, they are for reference only. The descriptions that follow each line should provide enough information for you to try different values and to see the effects.

So, line by line, we have:

#### 1 *double\_buffer yes*

You will almost certainly want to include this line and say yes, it reduces flickering when Conky is updating its display. The only other acceptable value here is obviously no.

#### 2 *own\_window yes*

yes or no, again say yes.

#### 3 *own\_window\_argb\_visual yes*

colors are described in values from 1-255 of the amount of red, green and blue they contain, three color channels – rgb. By selecting this option a fourth channel is added, the 'alpha' or 'a' channel, hence argb. The alpha channel describes the amount of opacity. A value of 0 for full transparency, 255 for full opacity. This will usually be set to yes. A compositing window manager is required for transparency.

#### 4 *own\_window\_class Conky*

This line can be omitted as Conky is the default window class but the ability to set it can be useful if you want to use some window manipulation tools.

#### 5 *own\_window\_hints skip\_taskbar*

This line accepts a string of options separated by commas (no spaces). These options define the appearance and behaviour of the actual Conky window. The available options and their effects are:

undecorated - remove the window decorations such as the close button and title bar

below - force Conky below other windows

above - allow Conky to be displayed above other windows

sticky - forces the window to be visible on all virtual desktops  
skip\_taskbar - don't show Conky on the taskbar  
skip\_pager - don't show Conky on the virtual desktop pager

## 6 *own\_window\_transparent no*

This sets or removes the transparency of the Conky window. It only works when compositing is enabled.

## 7 *own\_window\_type normal*

There are four types of windows that Conky can generate.

- normal - this is the default
- desktop - this has most of the window hints enabled
- panel - this produces windows that cling to the screen edge
- override - the window is not controlled by the window manager and provision must be made to accommodate this – not recommended unless you really know what you are doing.

## 8 *own\_window\_title Conky*

This is the title that is displayed on the title bar when the window is 'decorated' and transparency is disabled. The default is to display the system hostname.

## 9 *update\_interval 5.0*

This decides how often Conky should update its display with new information. The value is in Seconds (5 seconds here). Refreshing too often causes Conky to eat more system resources. The value can only be ascertained after running Conky with a revised configuration file. A bigger value is usually better. Ask yourself "How often do I really need to see new information?" This value depends upon:

- The type of data you are requesting
- The work required to get this information,
- Any data manipulation or formatting that is required prior to displaying the data.

## 10 alignment top\_right

Where the Conky window should be displayed. Options are:

- top\_right
- top\_left
- top\_middle
- bottom\_right

bottom\_left  
bottom\_middle

These may be abbreviated to: tr, tl, tm, br, bl, bm as appropriate.

## 11 *background no*

For most users this line can be omitted. Saying yes forks the process to the background when Conky is started.

## 12 *border\_width 1*

This is a border filled with the background color, or the default color specified in line 14, between the window edges and the displayed content. In most cases a small value will suffice. Borders are drawn if line 19 says yes.

## 13 *cpu\_avg\_samples 2*

When CPU data is requested what is shown is not the instantaneous value but an average over a number of readings. The more readings are taken the more accurate the result. For most practical purposes a low value is sufficient.

## 14 *gap\_x 5*

The gap, in pixels, between the chosen alignment edge (line 10) and the conky window in the x (left – right) direction;

## 15 *gap\_y 5*

as above in the y (up – down) direction

## 16 *default\_color white*

This determines the default color when no other is specified and the color for the border declared in line 19. The color can be the name of a color as defined by the X11 color scheme which is nicely displayed on this web site:

<http://www.graphviz.org/doc/info/colors.html>

Alternatively a hexadecimal value such as FFFFFFFF may be used (Note: no leading #). The GIMP color selector is useful for obtaining these values.

## 17 *default\_outline\_color white*

The outline referred to here is that decided upon in line 21. Colors are as above

### 18 *default\_shade\_color white*

The color of the text drop shadow if enabled in line 22. Colors are as above.

### 19 *draw\_borders no*

This is the border described in line 12

### 20 *draw\_graph\_borders yes*

This draws a rectangular border around the area in which the graph will be displayed.

### 21 *draw\_outline no*

This option outlines the text and graph borders in the color chosen in line 17

### 22 *draw\_shades no*

The shades referred to are text drop shadows.

### 23 *net\_avg\_samples 2*

This works as for the CPU samples described in line 13.

### 24 *no\_buffers yes*

The buffers referred to here are memory buffers. The kernel does not release memory immediately, some data is held ready for re-use rather than re-loading but that memory is always available to be used when required. Saying yes here will display only the memory that is actually in use. Saying no reports the total memory currently holding data even if that data is not currently active.

### 25 *out\_to\_console no*

It can be helpful to say yes here if things are not working quite right. The output that Conky is trying to display is then also printed on the console which can make it easier to see where the error is. Normally you would want to say no.

### 26 *out\_to\_stderr no*

This is similar to the line above and can be used as a further debugging aid

### 27 *stippled\_borders 0*

If you have enabled borders then this allows you to have them broken or dashed. The value is the length, in pixels, of the gaps and the solid portions. A value of 0 gives a continuous border. Leaving out this line has the same effect as a value of 0.

### 28 *uppercase no*

If yes then text is forced to upper-case.

### 29 *use\_xft yes*

This provides support for X freetype font rendering which is generally a good thing. If you don't understand the difference try saying no to see which you prefer. That's what I did and then decided to say yes.

### 30 *xftfont Sans:size=10*

The default font face and size. This can be overridden in the text section.

### 31 *own\_window\_colour 3b3b3e*

Here you can set the background color of the window if transparency is set to no. Colors are as described in line 16.

### 32 *color1 ghostwhite*

Colors can be predefined for use in the text section using color0 to color9. This is useful for easily changing the look throughout the script and can also improve readability of the script. Colors again as line 16

### 33 *color2 skyblue* As above

### 34 *color3 navajowhite3* As above

### 35 *color4 green2* As above

### 36 *# End of configuration section*

This is a comment and is ignored. Anything that follows a # is taken to be a comment and comments can appear on a line of their own or be added at the end of a command statement although you should add at least one space between the command and the comment to clearly separate the two.

There is plenty to play with here, and there are lots more in the documentation. These are the ones that I use the most.

### The TEXT Section

The word TEXT on its own line declares the start of the output section of the configuration file, and must follow the configuration section as described above. Case is not sensitive here, but I prefer to use upper-case to make it easier to see where this section begins.

What you have here is a way to display static text, use the contents of pre-defined variables (names that refer to a value that can change, such as the speed of a fan) to provide system relative information in either a numeric, textual or graphical format, or to show the current status of an on-going process, perhaps the play status of a music track. By 'static text' I mean text that is output exactly as typed and does not change according to the value of a variable or the returned value of a function. This is what a system monitor does, and this is exactly what Conky does – did I leave anything out? If I did, then Conky can probably do that, too.

Comments can be added in here, but a blank line will output a blank line. Spacing plays an important role in the text section, and so comments should be used with due care and consideration.

Each of Conky's variables should be preceded by a '\$' and it is better to enclose them and any arguments in curly braces, as shown below. The braces are not strictly necessary, but help to reduce spacing and parsing errors. Without the braces, text that follows a variable must be preceded by a space in order for Conky to be able to correctly recognize the variable. The space will then be printed, which is not always desirable. Without the space, Conky will not recognize the variable and will print out the entire string verbatim, if possible. Get into the habit of always doing this, and you will have fewer problems with your output.

Below are the contents of the text section for the demo system monitor, numbers added for reference are not part of the script:

Again, line by line, although some of the explanations may be a little longer here.

#### 37 TEXT

The section starts here

```
38 #{image /home/user/Pictures/conky/pcloslogo.png -p 27,0 -s 160x60}
```

All of the text that follows the line number should be on one line. It is possible to split the line with a backslash if you are careful but this can cause problems if the backslash is in an inappropriate place. I have all of the images I want to display in Conky in a folder which is inside the Pictures folder of my home directory. This line will place the named image in the window. The -p 27,0 positions the image 27 pixels right and 0 pixels down relative to the top left of the window. The -s 160x60 scales the image to 160 pixels wide by 60 pixels high.

```
39 #{voffset 0}
```

The value of 0 here does nothing but a larger value will instruct Conky to start printing its output at a position the corresponding number of pixels lower down. Similarly a negative value will move the insertion point upwards. The line is included here to allow me to control the layout more easily if I decide to make some changes.

```
40 #{font Liberation Sans:size=10}
```

This command will temporarily override the font settings given in the configuration section, line 30

```
41 #{color2}#{hr 1}
```

Here color2, defined in line 33 is used to draw a horizontal line with a height of 1 pixel. The color may also be given as a 6 character hexadecimal number with or without the leading #. The length of the line will be adjusted to fill the available space. Look at the bottom of the image above, where we have 'Mail' to see how this works.

```
42 #{color1}#{alignc}#{exec VERSION=`mate-session --version | awk '{ print $2 }'`; echo -n Mate Version $VERSION}
```

This line starts to get a little more complex. From the left.

Use pre-defined color1,

Align the output centrally in the window.

Execute the following shell command. The command above can be broken down like this: get the version of mate that is running which, on my current system, retrieves mate-session 1.10.0

Pass this to awk which discards the mate-session part and then stores the remainder, the 1.10.0 part, in the variable VERSION.

Finally, echo the text "Mate Version" followed by the contents of the variable

VERSION so that we see Mate Version 1.10.0 on the screen. Doing it this way the current version is always displayed. For other desktop environments the command would need to be adjusted accordingly. The `exec` variable is used to execute shell commands to produce the required output. Shell commands can be quite resource intensive the `exec` variable executes the code at every update interval (line 9) and therefore Conky provides four other variants of the variable, `execp`, `execi`, `texeci` and `execpi`.

The `execp` variable is provided for convenience, it allows you to mix other Conky variables in with your shell script and Conky will correctly interpret (parse) them but this is very resource intensive and you may have to increase the update interval to achieve acceptable operation. The `execi` variable adds an execution interval that must be greater than that defined in line 9 above. This allows you to control when the shell commands are executed independently of the conky update interval set in line 9. More useful is `texeci` which, like `execi` includes an interval but here the shell code is executed in its own thread, returning the data to Conky and destroying the thread after completion. See line 82 for an example of the use of `texeci`. Finally `execpi` is `execp` with an interval argument. To be effective the shell code must exit before the data can be returned to Conky so commands such as `tail -f`, which follows the additions to a file until interrupted, will not work.

```
43  ${color4} ${time %H:%M %p} ${alignr} ${time %d %b %Y}
```

Now we use a different pre-defined color, a shade of green, to display first the time and then, aligned to the right, the date. This is confusing because the conky command for date and time output is `time` but the format string is the same as the Linux `date` command. The output is suitable for my locale but users in the US might prefer to change the last time command to  `${time %b %d %Y}`. Typing `date --help` in a terminal will show all of the available options for time and date formatting.

```
44  ${image /home/user/Pictures/conky/summary.png -p 0,118 -s 22x22} ${color1} System Information ${color2} ${hr 1}
```

Another image scaled and positioned to suit, followed by the simple text 'System Information' there are 8 leading spaces and 1 trailing space used to align the text although they don't show in the line above. See line 55 for a better positioning method.

Following this we change to `color2` and use a horizontal line to fill in the remaining space.

```
45  ${font Liberation Sans:size=8} ${color2} Hostname  
 ${color3} ${alignr} ${nodename}
```

Another font override. This time only the size is changed to be slightly smaller.

A switch to `color2`, The static text "Hostname " (note the trailing space), Then change to `color3`, the text following to be aligned to the right right. This text is the value of the Conky variable  `${nodename}` which is the host name of the currently accessible machine, usually, but not necessarily, the machine that you are working on.

```
46  ${color2} Kernel ${color3} ${alignr} ${kernel}
```

Back to `color2`, the static text "Kernel ", `color3` once again and then the value of the Conky variable  `${kernel}` which shows the version of the kernel currently in use. This last value is aligned to the right of the Conky window.

```
47  ${color2} Uptime ${color3} ${alignr} ${uptime}
```

This line follows the format of the line above but shows how long the system has been up (time since last reboot).

```
48  ${color2} CPU Speed ${color3} ${alignr} ${freq} MHz
```

Again a similar output but this time showing the current (averaged) speed of the CPU.

```
49  ${color2} Fan Speed ${color3} ${alignr} ${exec sensors | sed -n  
 '/fan1/p' | tr:w -s " " | cut -d" " -f 2 } RPM
```

This time we have to use a little more shell stuff to display the information that we need. Start with `color2` Print the text 'Fan Speed ' to the Conky window. Change to `color3`, right align the next text which is retrieved by executing the following shell commands: `sensors` – Displays the output provided by `libsensors`, which is usually quite a lot, this is then piped to the `sed` command which isolates the line containing data for `fan1`, this data is then passed to the `tr` command which uses the `-s` option to 'squeeze' repeated spaces from its output. This is then passed to the `cut` command which splits up the data into groups of characters separated by a single space and outputs the second group, or field, of found characters. The final stage of this line prints out the static text ' RPM' which includes the separating space.

```
50  ${color2} CPU Temp ${color3} ${alignr} ${exec sensors | sed -n  
 '/(crit/p' | sed -n '/temp1/p' | cut -d" " -f 9 | sed 's/+//'
```

Now we need the CPU temperature and this is another complex command. What I do to get the correct output, stripped of the superfluous stuff, is to keep trying the command in a terminal until the results are exactly what I need. For the above `exec` command I opened a terminal and typed the command `sensors` This produced a screenful of output. After reading through this output I used a

sed command to look for keywords that will reduce the number of hits. After that it is a process of elimination and best guessing to determine the correct values and choose one.

Next the line is trimmed to contain only the required data. I do realise that this is a tortuous route to travel to get a simple bit of information but after doing it a couple of times it does become easier and you do learn a lot about using useful tools such as sed, awk and cut.

```
51  ${color2}GPU Temp ${color3}${alignr}${exec sensors | sed -n  
'/temp10/p' | cut -d" " -f 8 | sed 's/+//'} 
```

Similar to line 50

```
52  ${color2}HDD Temp /dev/sda ${color3}${alignr}${exec hddtemp  
/dev/sda | awk '{print $4}'} 
```

Similar to line 50

```
53  ${color2}HDD Temp /dev/sdb ${color3}${alignr}${exec hddtemp  
/dev/sdb | awk '{print $4}'} 
```

Similar to line 50

```
54  ${color2}Battery Temp ${color3}${alignr}${exec sensors | sed -n  
'/temp5/p' | cut -d" " -f 9 | sed 's/+//'} 
```

Similar to line 50

```
55  ${color2}${if_match "${battery_percent}"=="0"}Battery  
%${alignr}${color3}Battery not installed${else}Battery  
Charge${goto 100}${color2}${battery_bar  
8,90}${alignr}${color3}${battery_percent}%${endif} 
```

This time we use a conditional expression: We want to change the output according to the battery conditions that exist. If there is no battery installed then we state the fact and exit the line. If the battery shows a charge then it must be installed. Now we display a graphical bar gauge and the percentage charge. First we set the color to color2. Next we read the value of the Conky variable `{battery_percent}` and test the value we read against the absolute value "0" with the `{if_match "${battery_percent}"=="0"}` Conky test. If we get a match then we print "Battery %" and then, right aligned, print the text "Battery not installed" then we move on to the next line. If the test fails, then there is a charge detected and a battery must be present. Now the `{else}` statement kicks in and we print "Battery charge" move the insertion point to 100 pixels right with `{goto 100}`. Inserting spaces as we did in line 44 is one way to

position text but has the disadvantage that it is dependent on the size of the font, images and other components that make up the line. Using the `{goto}` Conky variable positions the insertion point a fixed number of pixels from the left hand window edge, in this case 100 pixels. This will provide a consistent look to the output. change to color2, print a battery bar gauge 8 pixels high, 90 long. Next, right aligned and in color3, we print the value of the Conky variable `{battery_percent}` followed by a % sign. We finally end the conditional expression with the Conky variable `{endif}`.

```
56  ${color2}Procs / Running${color3}${alignr}${processes} /  
${running_processes} 
```

Set the color to color2, print static text 'Procs / Running', change to color3, change alignment and print the value of some Conky variables separated by '/'

```
57  ${color2}Load Average ${color3}${alignr}$loadavg 
```

color, text, color, alignment, Conky variable value

```
58  ${font Liberation Sans:size=9}${color1}${image  
/home/user/Pictures/conky/fft.png -p 0,336 -s 30x20}${goto 46}CPU  
Use ${color2}${hr 1} 
```

This is a header line as line 44 but uses the `{goto}` variable to position the insertion point.

```
59  ${font Liberation Sans:size=8}${color2}Core 1  
${color3}${alignr}${cpu cpu1}% 
```

Change the font size and use color1."Core 1" is static text and is simply printed out. Change output color to color3, use right alignment and output the percentage of core 1. Individual cores are identified as `cpu1`, `cpu2` etc. the argument `cpu0` is used to output the total cpu usage. The final "%" is static text.

```
60  ${color2}${cpubar cpu1 8} 
```

Output a cpu usage bar gauge 8 pixels high for core 1

```
61  ${color2}Core 2 ${color3}${alignr}${cpu cpu2}% 
```

As line 60 but using core2

```
62  ${color2}${cpubar cpu2 8} 
```

Output a cpu usage bar gauge 8 pixels high for core 2

```
63 font Liberation Sans:size=9color1image
/home/user/Pictures/conky/ram-use.png -p 0,410 -s 34x26goto
46Memory Use color2hr 1}
```

This is a header line as line 58.

```
64 font Liberation Sans:size=8color2Memory
color3alignrmem / memmax (memperc%)
```

Resize font again, change color to color2, insert the static text 'Memory', change to color3 and right alignment, insert the current value of memory usage (this is affected by line 24), static text '\ ', the total amount of memory available to the system, static text ' ' (a space) and the percentage of memory in use (again affected by line 24). This does not include swap space, there are separate Conky variables available for that.

```
65 color2membar 8}
```

Display an 8 pixel high memory usage bar graph in color2.

```
66 font Liberation Sans:size=9image
/home/user/Pictures/conky/hdd.png -p 0,462 -s
20x20color1goto 46File System color2hr 1}
```

This is a header line as line 58.

```
67 color2rootgoto 60hascolor3alignr fs_free / free
```

I have three internal drive partitions that I want to monitor and they are mounted at '/' (the root partition), '/home' and at '/home/user/data'. The root and home partitions are always mounted and so their lines are a simple report using the `fs_free` Conky variable that shows the remaining free space on the partition.

```
68 color2/homegoto 60has color3alignr fs_free /home
free
```

As line 67

```
69 color2if_mounted /home/user/data~/datagoto 60has
color3alignr fs_free /home/user/data
freeelse~/datagoto 60iscolor3alignrnot mountedendif
```

Sometimes this partition is not mounted to protect the data it holds and so as

well as knowing the available free space on the device I want to also show the mount state. For this I need to use a conditional statement.

First I test the `if_mounted` Conky variable. If this test returns positive (true) then everything up to the `else` statement is output and execution of the line is complete. If the test fails (false) then execution jumps to a point immediately after the `else` Conky variable and execution continues outputting the text in appropriate colors and right aligned.

```
70 image /home/user/Pictures/conky/ext-hdd.png -p 0,530 -s
22x22color1goto 46Removable Drives color2hr 1}
```

This is a header line as line 58.

```
71 font Liberation Sans:size=8color2if_mounted
mnt/backupbackupgoto 60has color3alignr fs_free
mnt/backup freeelseif_existing /dev/disk/by-
label/backupbackupgoto 60iscolor3alignrnot
mountedelsebackupgoto 60isalignrcolor3not
connectedendifendif
```

I also have two external drives that I want to monitor. I use labels to identify my partitions but I could also use device nodes such as `/dev/sda1` or the devices UUID. My drives are labelled 'backup' and 'share'. In this case each of the drives can be in one of three states:

- Mounted
- Connected but not mounted
- Not connected.

This line checks first if the drive is mounted using the Conky variable `if_mounted` and if so, reports accordingly and the line is complete. If the drive is not mounted then execution of the line skips to the code after the `else` Conky variable and attempts to locate the drive using the `if_existing` Conky variable. Linux provides links to the drives and partitions that it knows about in the `/dev` system directory. The three places that we should look are:

- `/dev/disk/by-label`
- `/dev/disk/by path` (device node)
- `/dev/disk/by-uuid`.

If the test - `if_existing /dev/disk/by-label/backup` finds the partition then it must be in the second state – 'connected but not mounted' and this fact is reported and the line execution terminates. If the test does not find the drive then it is apparent that the drive is in the third state – 'not connected'. This is reported and the line execution is complete.

```
72 {color2}{if_mounted /mnt/share}share{goto 60}has
{color3}{alignr} {fs_free /mnt/share} free{else}{if_existing
/dev/disk/by-label/share}share{goto 60}is{color3}{alignr}not
mounted{else}share{goto 60}is{alignr}{color3}not
connected{endif}{endif}
```

As line 71

```
73 {font Liberation Sans:size=9}{image
/home/user/Pictures/conky/processor.png -p 0,580 -s
30x20}{color1}{goto 46}CPU % {color2}{hr 1}
```

This is a header line as line 58.

```
74 {font Liberation Sans:size=8}{color2}{top name 1}
{color3}{alignr}{top cpu 1}
```

The `{top}` conky variable reports on processes by CPU usage ranked from highest to lowest. The variable takes 2 arguments, type and ranking. Here we report by type name and ranking 1 (the highest CPU usage).

Type can be any one of: name, pid, cpu, mem, mem\_res, mem\_vsize, time, uid, user, io\_perc, io\_read or io\_write.

The top 10 processes may be reported on.

```
75 {color2}{top name 2} {color3}{alignr}{top cpu 2}
```

As line 74 for process ranked 2

```
76 {color2}{top name 3} {color3}{alignr}{top cpu 3}
```

As line 74 for process ranked 3

```
77 {font Liberation Sans:size=9}{image
/home/user/Pictures/conky/memory.png -p 0,644 -s
30x20}{color1}{goto 46}Memory % {color2}{hr 1}
```

This is a header line as line 58.

```
78 {font Liberation Sans:size=8}{color2}{top_mem name
1}{color3}{alignr}{top_mem mem 1}
```

As line 74 for process ranked 1 but sorted for memory usage instead of CPU usage.

```
79 {color2}{top_mem name 2}{color3}{alignr}{top_mem mem 2}
```

As line 74 for process ranked 2 but sorted for memory usage instead of CPU usage.

```
80 {color2}{top_mem name 3}{color3}{alignr}{top_mem mem 3}
```

As line 74 for process ranked 3 but sorted for memory usage instead of CPU usage.

```
81 {font Liberation Sans:size=9}{image
/home/user/Pictures/conky/network.png -p 0,709 -s
20x20}{color1}{goto 46}Network {color2}{hr 1}
```

This is a header line as line 58.

```
82 {font Liberation Sans:size=8}{color2}Bluetooth{image
/home/user/Pictures/conky/bluetooth.png -p 60,724 -s
18x18}{if_match "{texeci 10 cat /proc/acpi/ibm/bluetooth | grep
status | awk '{print
$2}'"=="enabled"}{alignr}{color3}enabled{else}{alignr}{color
3}disabled{endif}
```

My laptop is a Thinkpad and the status of the bluetooth device is stored in the file `/proc/acpi/ibm/bluetooth`. For other machines there will probably be a similar entry. The bluetooth status can be either 'enabled' or 'disabled'. This line uses the `{if_match}` Conky variable to determine the current status and report on it. The search is done using the `{texeci}` Conky variable which starts a new thread to execute the code and repeats it at 10 second intervals. This allows the rest of Conky to go about its business while this low priority search is in progress. The rest of the code in the line should be familiar by now.

```
83 {if_up wlan0}{color2}Wireless{image
/home/user/Pictures/conky/network-wireless.png -p 61,740 -s
18x18}{alignr}Signal Strength
{color3}{wireless_link_qual}%{else}{color2}Wireless{image
/home/user/Pictures/conky/network-wireless.png -p 61,740 -s
18x18}{endif}
```

This line checks the wireless network adaptor, `wlan0` for a connection and if found reports the signal strength. The check can be made more strict by an entry in the configuration section. The `{if_up_strictness}` Conky variable accepts an argument that can be:

- up – but not necessarily with a link
- link – up and with a link

address – up, with a link and an associated IP address

If the adaptor is not connected then only the image and the word 'Wireless' are output.

```
84 Down Downspeed wlan0/s AlignUp  
Upspeed wlan0/s
```

Still with the wireless adaptor, this reports on the download and upload speeds

```
85 Downspeedgraph wlan0 25,107 7ac6f7 d7ba8e -l -t  
Upspeedgraph wlan0 25,107 7ac6f7 d7ba8e -l -t
```

Displays a graph of the download speed of the wireless adaptor, 25 pixels high and 107 pixels wide using 2 different colors defined in hexadecimal with no leading #. The -l argument uses a logarithmic scale enabling small values to be displayed and the -t argument introduces a temperature gradient changing color according to the value recorded.

```
86 Total Totaldown wlan0  
Total Totalup wlan0
```

The totals downloaded and uploaded by the wireless adaptor.

```
87 EthernetImage Image /home/user/Pictures/conky/ethernet.png  
Image -p 60,826 -s 20x20}
```

The start of the output for the Ethernet adaptor eth0

```
88 Down Downspeed eth0/s AlignUp  
Upspeed eth0/s
```

As line 84 but for eth0

```
89 Downspeedgraph eth0 25,107 7ac6f7 d7ba8e -l -t  
Upspeedgraph eth0 25,107 7ac6f7 d7ba8e -l -t
```

As line 85 but for eth0

```
90 Total Totaldown eth0  
Total Totalup eth0
```

As line 86 but for eth0

```
91 Font Liberation Sans:size=9Image  
/home/user/Pictures/conky/ip-addresses.png -p 0,911 -s  
22x22}Color1Goto 46IP Adresses Color2Hr 1
```

This is a header line as line 58.

```
92 Internal IP wlan0: AlignColor3Addr wlan0
```

The IP address assigned to wlan0

```
93 Internal IP eth0: AlignColor3Addr eth0
```

The IP address assigned to eth0

```
94 Gateway IP:AlignColor3 Gw_ip
```

The gateway IP address

```
95 External IP: AlignColor3Execi 3600 wget  
http://checkip.dyndns.org -0 - -o /dev/null | cut -d : -f 2 | cut  
-d \< -f 1}
```

The external IP address downloaded from an external website at one hourly intervals. For security reasons this value should not be openly displayed and so I have removed it in the image above.

```
96 Font Liberation Sans:size=9Image  
/home/user/Pictures/conky/mail.png -p 0,998 -s  
20x20}Color1Goto 46Mail Color2Hr 1
```

This is a header line as line 58.

```
97 Font Liberation Sans:size=8Color2Unread Internal Mail  
AlignColor3 New_mails /var/spool/mail/user}
```

Check for new internal e-mails. This feature must be set up to be used here. I use postfix so that I can receive notifications from services like cron and from backup scripts.

### Using Conky

To try out Conky, create a configuration file in your home directory and add all of the configuration section above, lines 1 to 36 inclusive. Follow this with the word

## Monitor Your System With Conky, Part One

TEXT on its own line. Save the file with a name such as conky-template. This template will work for most of the Conky configurations that you need to create, with only minor changes.

Next add a single, complete line from the text section, that's lines 38 to 97. It doesn't matter which line (these are only examples), but any lines that display images must be adjusted to point to an actual image, and any partitions or network interfaces must be similarly adjusted. Choose a simple line to start with, and don't try to add too many lines yet, or you may have problems if things don't work correctly. Re-save the file with a suitable name, e.g. `~/.conkyrc_mag`.

Open a terminal and call the file you created with `conky -c ~/.conkyrc_mag` (or whatever you named your file, and don't forget the leading period).

If all goes well, you should see the output from that line. If not, then check your typing. If you need to make any corrections and you saved a template file then, you should make the same changes to that file. Do it now before you forget them. Once you have this running, you can add and remove lines in the TEXT section, and change things around to build your own, personalized Conky. Try changing some of the values in the configuration section to see their effect. Use a different font or several fonts. Certainly change the color values to suit your theme or mood. This is a personal thing, and you will never be happy with your work until it looks right for you.



If you followed so far, then you will have a good grasp of Conky configuration. The script I used as an example suits my needs, and covers most of the basics. Here are a couple of examples to give you ideas. You should know enough by now to be able to produce something similar.



If you look on the internet, you will find many examples like these, and some a lot more

advanced than these. Some are so advanced that Conky needs assistance to be able to show them, and we'll cover all of that in due course. First let's look at a few more things that the basic Conky can achieve.

Conky is not limited to reporting purely system related information. It can be quite a versatile tool. For example, in an article I wrote for the magazine about the Music Player Daemon, mpd, I demonstrated the use of Conky to display the name and status of the track playing and an image of the album cover. The configuration file is a part of the article, so I won't repeat it here, but the Conky display looked like this.



### RSS

To display an RSS feed in a Conky window, open your template file and resave it as `~/.conkyrc-rss`.

In the configuration section:

change `own_window_title conky` to `own_window_title Breaking News`

Add `minimum_size 400 250` and `maximum_width 400`

This will give you a fixed size display even when the window contents change.

Your template file should have been saved with an empty TEXT section so in the TEXT section add just these lines:

```
_${color2}${rss http://feeds.bbc.co.uk/news/world/rss.xml  
30 feed_title}
```

```
{color1}{rss http://feeds.bbc1.co.uk/news/world/rss.xml  
30 item_titles 10 4}
```

That is all that is needed to create a new Conky once you have the basic template. Before I show the results I'll explain what each change does.

The **{rss}** conky variable accepts the following arguments: uri, interval, action, num\_par and spaces in front. The last two are optional.

**uri** is the address of the rss feed, here this uses the http protocol, but this Conky variable should be OK with any protocol that the curl command can handle.

**Interval** is the update interval in minutes

**action** refers to what part of the feed you want to display. This can be:

feed_title	The title of the rss feed
item_title num_par	An item heading numbered from 0 up
item_desc num_par	The text from the numbered item
item_titles	A list of item titles numbered from 0 to num_par. With this action the option argument spaces_in_front will indent the items by that many spaces

**num\_par** is the item number from the feed

This rss Conky variable runs in its own thread and Conky must be re-started to correctly show any changes to the configuration file.



The text from each line supplied by the item\_desc argument can be quite long, and Conky is not very good at wrapping the text. The information displayed here is enough to notify me of anything I might like to investigate further. Although

I have demonstrated the feature as a stand-alone application, the rss content can form part of a more complex Conky display.

## Transparency

We can easily make this Conky display transparent by changing two lines in the configuration section. For this to work, compositing, or desktop effects, must be enabled.

Change

```
own_window_transparent no
```

to

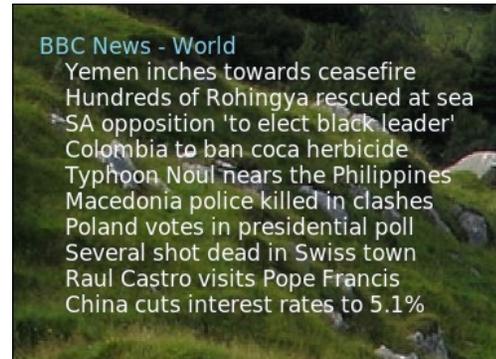
```
own_window_transparent yes
```

and

```
own_window_hints skip_taskbar
```

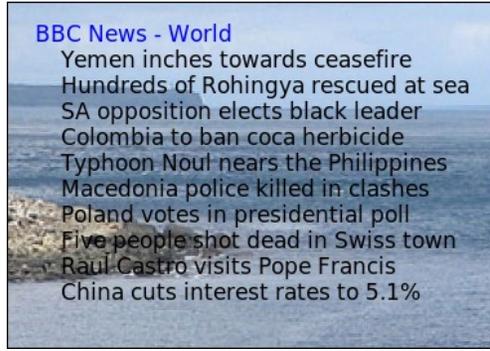
to

```
own_window_hints skip_taskbar,undecorated
```



This demonstrates one of the difficulties with transparency. The two images above are of the same Conky display on two different areas of the same background image. The colors could, of course, be changed to accommodate this, but this is something that should be considered when designing a Conky display.

The next two images show the same thing, first with compositing turned off and then with compositing back on, but with the text colors changed to contrast better with the background image. If you see image three, check you have enabled compositing!



## Templates

The use of templates can simplify configuration files, particularly when the same type of information is required over several lines.

Line 67 of .conkyrc\_mag had this line:

```
 ${color2}root${goto 60}has${color3}${alignr} ${fs_free /} free
```

Line 68 was almost identical.

If I added a template definition to the configuration section:

```
template1 ${color2}\1${goto 60}has${alignr}${color3}${fs_free \2}
```

Then lines 67 and 68 would become:

```
 ${template1 root /}
 ${template1 home /home}
```

Line 69 was

```
 ${color2}${if_mounted /home/user/data}~/data${goto 60}has
 ${color3}${alignr} ${fs_free /home/user/data}
 free${else}~/data${goto 60}is${color3}${alignr}not mounted${endif}
```

This gets simplified to

```
 ${if_mounted /home/pete/data}${template1 ~/data
 /home/pete/data}${else}~/data${goto 60}is${color3}${alignr}not
 mounted${endif}
```

Putting a lot of the complexity out of the main body of the TEXT section which makes things easier to control.

This is how the templates work.

A template is defined/described in the configuration section of the configuration/resource file. Yes, I know the terminology is confusing, just run with it and soon you'll be fine. A template definition has a name comprising the label 'template' followed immediately by a number from 0 to 9. Our template is named 'template1'. The rest of the template definition is constructed just like a line from the TEXT section of the file. The exception here is that where in a TEXT section line we would insert the names of the items that we want Conky to report upon, the template has a 'place holder' which is a number preceded by a backslash. I think the numbers can be from 1-9, but I'm not sure – use 1-9, you'll never need more than 1-4 anyway! Place holders are analogous to place holders in extended regular expressions for those who understand that concept.

When a template name is encountered in the TEXT section of the file it is substituted by the entire contents of the template definition with a few minor changes. These changes are that the placeholders are replaced by the values that immediately follow the template name.

In our example this means that \1 is replaced by the first item following the template call, \2 by the second item etc. So if \1 is replaced by root and \2 is replaced by / then

```
 ${template1 root /}
```

becomes

```
 ${color2}root${goto 60}has${alignr}${color3}${fs_free /}
```

and

```
 ${template1 home /home}
```

becomes

```
 ${color2}home${goto 60}has${alignr}${color3}${fs_free /home}
```

The placeholders are replaced by the items that immediately follow the template call. The rest of the items in the line e.g `$(goto 60)` are interpreted as normal.

If this seems confusing, and it certainly was at first to me, then try a few examples and I am sure that you get a grasp of the concept.

## Gauges and Bars

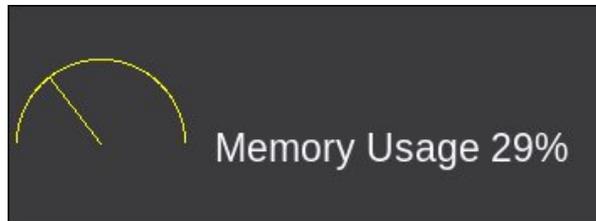
We've seen bars already, battery bars, memory bars and cpu bars. They give an instant visual impression of the usage of the item that is being represented. A battery bar shows how charged the battery is, not by a number but by the length of the bar proportional to the possible maximum length of the bar. The color can be changed, as well as the length. The height can be changed and the position of the bar within the display, but nothing else. They are useful but not really very interesting.

Conky has another type of proportional representation the we haven't yet seen – the elliptical gauge, which works in the same manner as a pressure gauge or a speedometer. Gauges are available for all of the things that bars can be used to represent, battery charge, memory usage...

Using these two lines

```
${font Liberation Sans:size=14}
${color yellow}${memgauge 50,100}${color1} Memory Usage
${memperc}%
```

produces this output:



Okay, it works and you can see what you are meant to see. In some situations this is just fine and no more is required. But hey, this is Linux, and we don't settle for ok. We want WOW!

If you want WOW!, we can do WOW! But you've got to work for it. You need Lua and Cairo. We will cover that in Part Two.

## The PCLinuxOS Magazine Special Editions!

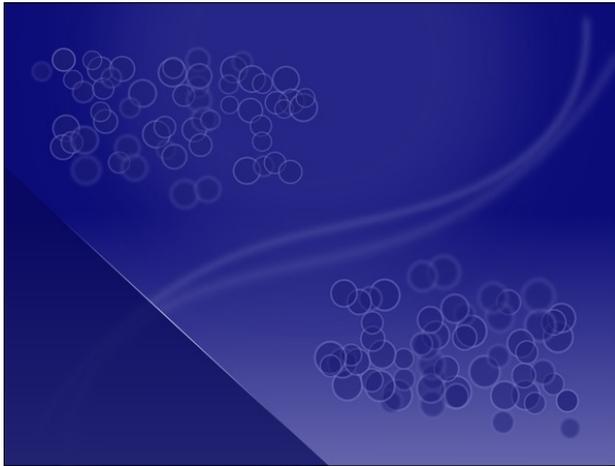
**Get Your Free Copies Today!**



# Inkscape Tutorial: Abstract Wallpaper

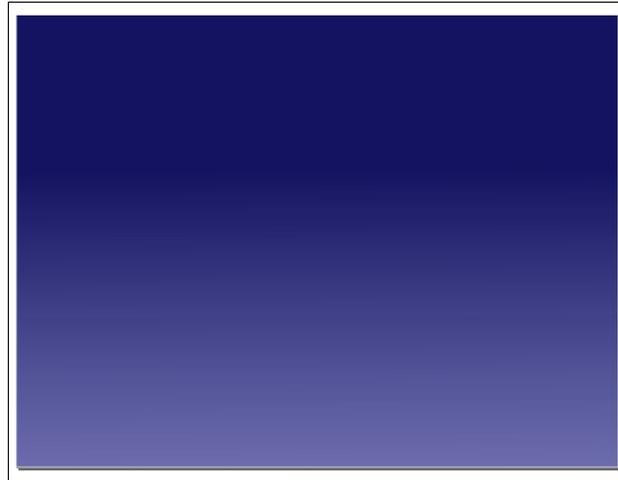
by Meemaw

Many of the wallpapers we see have circles that look like they are different sizes, but all look similar. We have now worked with clones enough to know that the **Tiled Clones** tool can do that pretty easily. I saw this tutorial not long ago, and thought it looked pretty fun, so let's play a bit.

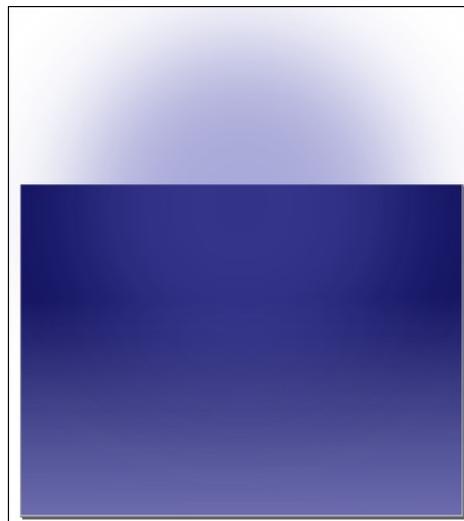


Open Inkscape and edit your **Document Properties**. My page will be 1200 X 900, but yours can be whatever you want. Start with a rectangle the same size as your page, and make it whatever color you want. My preferred color is blue, and I'm going to make it a bit dark to start. When you have your color in, change the stroke to none, and the fill to a linear gradient, with your original color on the top of the page, and a lighter shade of that color on the bottom (center, top).

Choose the circle tool and draw a circle with fill the same color as the bottom of the gradient (or at least a color that blends well with it) and no stroke. Make the circle about two-thirds as wide as your page. My



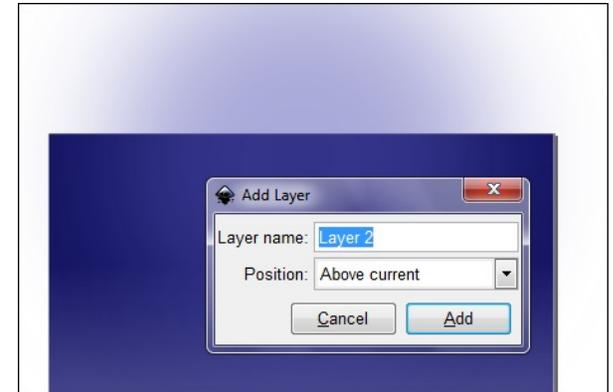
page is 1200 pixels wide, so I made my circle 800 pixels. Remember, after you get your circle drawn, you can adjust the size in the boxes along the top of your workspace. Put it about halfway off the top of the page, and center it horizontally using the **Align and Distribute** tool. Finally, change the Blur to about 25% and the Opacity to about 50% (but you



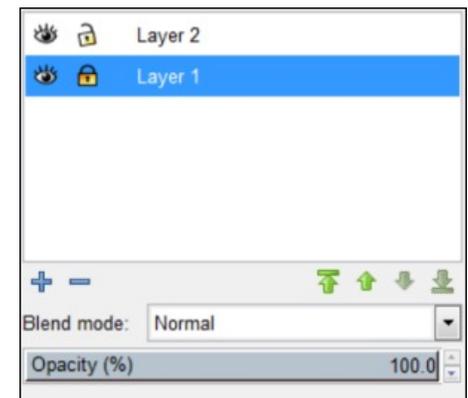
are free to play with these numbers and choose what looks good to you - we are just looking for a highlight at the top of our page).

Also remember that when you export your wallpaper, you will choose **Export Page**, and the part of the circle not on the page will be omitted.

Next, let's add a layer to our drawing. I want to be able to put more objects in without moving or changing anything I have already drawn. That's the great thing about layers. Generally we add a layer above the one we have been working on, so when you choose the **Add Layer** tool, the pop-up that appears has that as the default. Click **Add**.

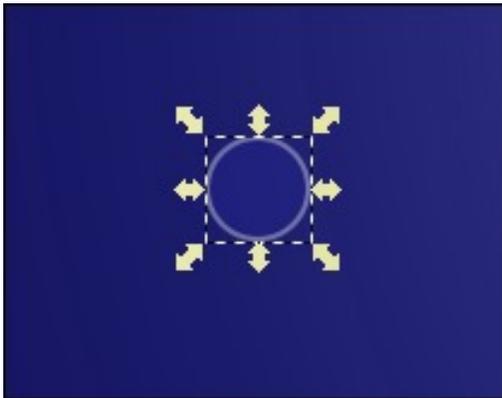


As an extra safety precaution, you can go back to Layer 1 and lock it by clicking the little lock on that line.



If for some reason you added the new layer below the original, you can move your rectangle and circle to the bottom by selecting them both and then choosing **Layer > Move Selection to Layer Below** or pressing **<Shift> + <Page Down>**.

Now let's add more on the top layer. Make sure that layer is selected, then draw a circle that is about 40 or 50 pixels wide. Make the fill one color, and a 3-pixel stroke using a lighter shade of the fill color. All of mine will be different shades of the blue I started using on the background. Change the opacity on your circle to 50%.

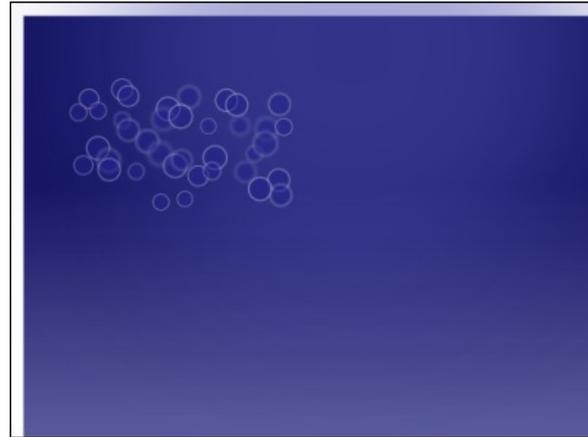


Let's multiply this circle by creating clones of it. Select it, then click **Edit > Clone > Create Tiled Clones**. The window that appears has several tabs and can be quite daunting. The best way to figure out what it does is to play with the settings and see what happens. For this tutorial, I got some help and will give you the settings I used.

When we did the stars in our sunset project, we put in rows and columns of clones. This time we will designate an area that we want to cover. On my 1200 x 900 pixel drawing, I used a 400 px width and 200 px height. Then the following tabs were set as shown below:

**Shift** - 50% in Randomize x and y,  
**Scale** - 25% in Randomize x and y,  
**Blur and Opacity** - Both 15% in Randomize. The Shift tab is shown at center top.

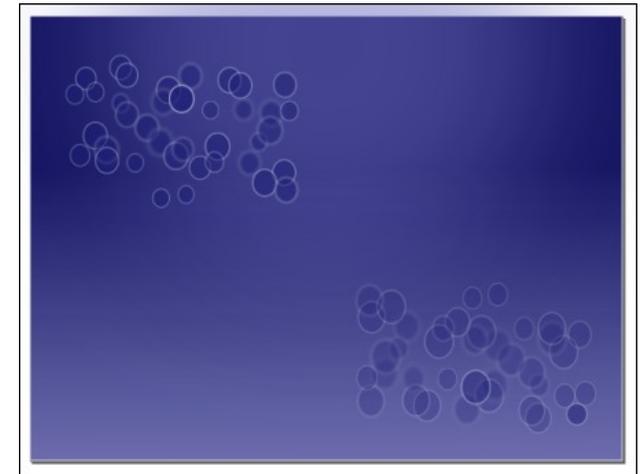
When you get all the settings in, click the **Create** button at bottom right. This is another place you can experiment: if you don't like the array that appears, you can click **Remove** and click **Create** again until you see something that's more pleasing to you.



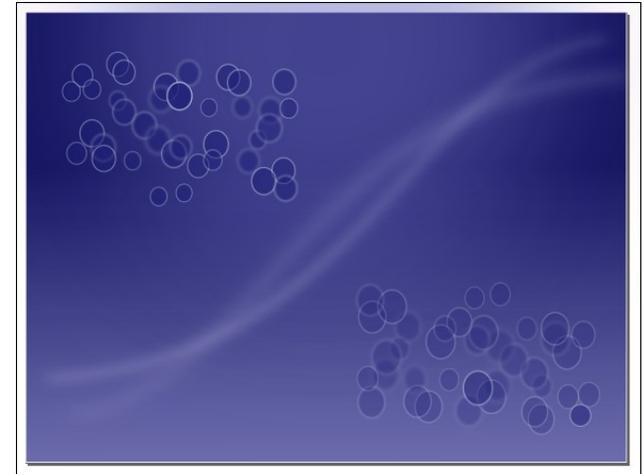
Wow, I've decided that I need to change the fill color of my objects! Remember, if you change the original, all the clones will be changed. OK, which one of those is my original? None are selected! This is cool.... choose one of the clones, then press **<Shift> + <D>**, and your selection will change to the original item. Now you can edit the fill. If you haven't saved your file yet, this is a great time to do so.

This is the basic tutorial, but I'm going to do more. First I will select all my clones, then duplicate the whole bunch of them (**<CTRL> + <D>**). Then I'm going to move them into another part of my drawing. I also want to flip mine and change their opacity so they don't look identical to the others. I also increased the size of both groups. Increase the original first, because the change to the original circle will affect them all since they are all clones. Then if you want your duplicates bigger or smaller again, select the second group and change them (top right).

The next thing I am going to do is put a couple of swirls through the picture. I will use the pencil tool and make the stroke one of the lighter colors we are



using. I also changed the **Opacity** to 75% and the **Blur** to 3%. After I get a line I like, I will duplicate it and flip it so I have two lines. Since I only duplicated the first line to get the second line, I can change the Opacity and Blur to something different on the first without altering the second.



Another thing that you can do is place a rectangle or triangle on your wallpaper. My triangle will be in the bottom left of the page, and will be the darkest of the colors. I also positioned a line along one side of the triangle, and made it much lighter with a radial gradient as an accent (next page, top left).



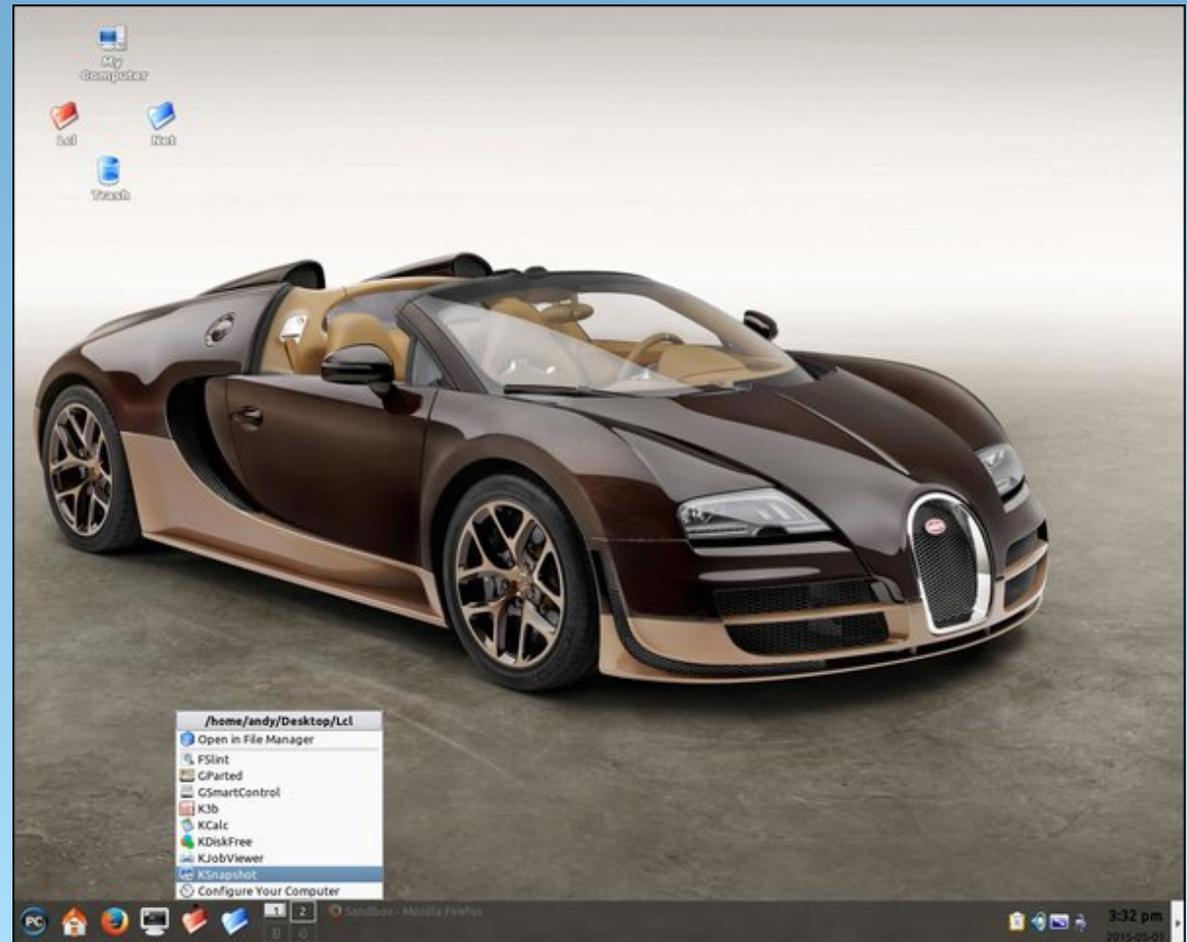
# Screenshot Showcase

You can do much more or much less than I have done. It's your project, so make it the way you want it.



**PCLinuxOS**  
Radically Simple

It's easier than  $E=mc^2$   
It's elemental  
It's light years ahead  
It's a wise choice  
It's Radically Simple  
It's ...



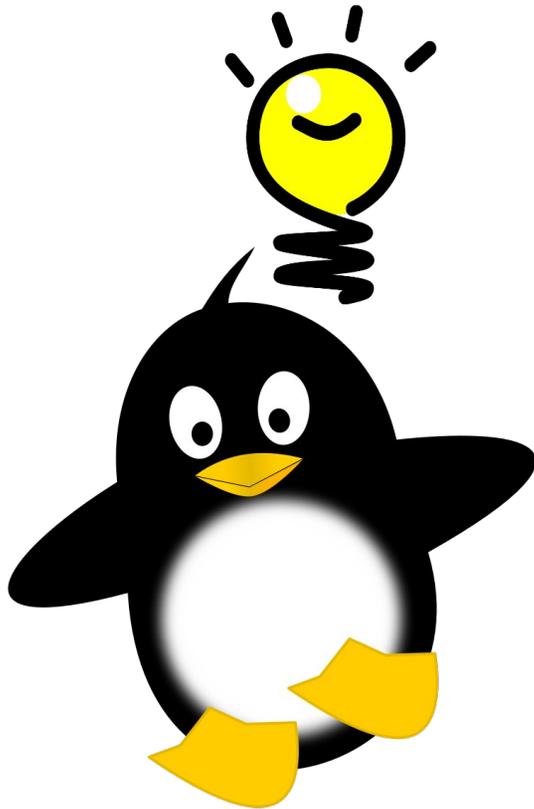
Posted by Andy Axnot, May 1, 2015, running Trinity.



# Tip Top Tips: Samsung Printer Driver Installer

**Editor's Note:** *Tip Top Tips* is a new monthly column in *The PCLinuxOS Magazine*. Each month, we will feature – and possibly even expand upon – one tip from the PCLinuxOS forum. The magazine will not accept independent tip submissions specifically intended for inclusion in the Tip Top Tips column. Rather, if you have a tip, share it in the PCLinuxOS forum's "Tips & Tricks" section. Your tip just may be selected for publication in *The PCLinuxOS Magazine*.

This month's tip comes from PCLinuxOS forum member Phil.



Try to set up your Samsung Printer via Configure Your Computer or through your browser localhost:631. It probably will not work for now.

Go to your local [samsung.com](http://www.samsung.com) website, say:

<http://www.samsung.com/us/support/downloads>

Search for your printer and look for the Downloads section, specifically for:

Print Driver (Driver) ver.V1.00.27.04 - Linux (MULTI LANGUAGE, 15.44 MB)

(CARE - by the time you read this it may well be a newer version)

gczerw says:

Extract the ULD\_V1.00.27.04.tar.gz file into a directory.

Using your file manager, switch into that directory and make sure that the \*.sh files are executable. If not, make them so.

Open a terminal window as root, change to the directory containing the driver files and do the following:

```
./install.sh [press Enter]
```

Then follow the prompts.

Job done.

\*\*\*\*\*

Samsung has an excellent history supporting Linux, at least with their printers. Six years ago, starting in 2009, Samsung came out with their Unified Printer

Driver for Linux. It now supports over 170 "series" of Samsung printers, and that number is only going up with the more printer "series" Samsung releases.



Their support of Linux was probably the most important key reason why I chose a Samsung printer to replace my HP LaserJet 1006, which died a premature death when the USB port burned out. I replaced it with the model ML-2525W, a small desktop laser printer (pictured above). Not only did it allow connection via USB port, but it also allowed connection to my router via ethernet cable and provided wireless 802.11 b/g printing capabilities. So, unlike my HP LaserJet 1006, I had three ways to connect it, as opposed to only the USB port.

Also key in my decision was the cost of toner cartridges. While not as inexpensive as the toner cartridges for Brother printers (Brother has some of the least expensive toner and ink cartridges in the industry), toner cartridges for the Samsung laser printers remain in the affordable range that won't break the bank ... or your wallet. Of course, the price of the printer also weighed heavily. I paid just under \$100 (U.S.) for the Samsung ML-2525W, which

astonished me, given the good reviews this printer received.

I do recall having a tough decision to make when I bought this printer, weighing it against a similarly featured and similarly priced Brother printer. While either would have fulfilled my needs, I don't regret the decision to purchase the Samsung. Having just had to "bury" an HP printer prematurely, I wasn't real keen on purchasing another HP printer – even though they do have a good reputation, both in Linux support and making quality products.

While Phil's instructions above are accurate, it might not be quickly apparent that a driver even exists when you visit the Samsung support website. So, let's illuminate exactly what you need to do when you get there.

Only the download pointed out with the red arrow is necessary. However, you may also want to download the two utilities pointed out with the green arrows, too. They may make administration of your Samsung printer easier. I have all of these installed on my computer, and I think that they really do help with printer administration. The same process that Phil points out for installing the Unified Printer Driver (red arrow) works also for the other two utilities.

an opportunity to support Samsung by purchasing and using their products. Rewarding companies for supporting Linux – with our purchasing power – is the only way that those "other" companies lacking Linux support for their products will realize that they are missing out on a market segment that has historically been underserved. In today's economy, every missed sale represents lost profits, and that is the only language that ALL companies seem to speak and understand.

### Summary

Since Samsung has made an honorable and concerted effort to support Linux, Linux users have



Downloads

Upgrade File (Firmware) ver.V1.xx.00.88 - All OS (MULTI LANGUAGE, 5.86 MB)	2011-okt-28
--	-------------

[DOWNLOAD \(ZIP\)](#) ↓

[See More +](#)

Under the "Downloads" section, click on the "See More +" link. The list should expand to show everything that is available, as in the excerpted image below.

Smart Panel (Driver) ver.V2.00.72 - Linux (MULTI LANGUAGE, 15.46 MB)	2011-aug-10
<a href="#">DOWNLOAD (GZ)</a> ↓	←
Unified Driver (Driver) ver.V1.00.06 - Linux (MULTI LANGUAGE, 15.35 MB)	2013-nov-11
<a href="#">DOWNLOAD (GZ)</a> ↓	←
Printer setting Utility (Driver) ver.V2.00.23 - Linux (MULTI LANGUAGE, 10.64 MB)	2009-nov-03
<a href="#">DOWNLOAD (GZ)</a> ↓	←

## Screenshot Showcase



Posted by ff103, May 1, 2015, running KDE.

# Leveraging MPD (Music Player Daemon)

---

by Peter Kelly (critter)

About six years ago, a few paragraphs in a PCLinuxOS Magazine article were devoted to describe the powerful, but low on resource usage, Music Player Daemon. You can find the article [here](#). The reason that the author skipped so lightly over it is, as he states in the article, that there is really not very much to say about it. It plays music. But it is a 'roll up your sleeves and get stucked in' sort of set up that not many users would want to be bothered with.

This music player can be controlled locally or over the network through clients, will use multiple outputs if required, has the ability to decode many types of music files, and maintains a database of your music files. In line with the Unix/Linux philosophy, it aims to 'do one thing and to do it well.' It does not, therefore, do video – it plays music.

Although the setting up of MPD is not difficult, neither is it obvious or particularly straight forward, particularly for users not used to having to roll up their sleeves to get something working. This raises a few questions.

**\* Why bring up the subject now?** Well, in six years a lot has happened in the Linux world that makes this utility much easier to integrate into a Linux/Unix environment.

**\* Why would anybody want the hassle of setting up such a system?** The big advantage of MPD is that it is so lightweight that it will happily play music on the most humble of hardware without getting in the way of using the computer for other, less frivolous purposes. Its memory requirements and CPU usage are barely noticeable.

**\* Are there any other advantages over easier to use, feature-rich graphical applications like Amarok, Clementine or Deadbeef?** Most dedicated music players, when installed, bring with them a whole host of libraries and files which they depend upon in order to function. MPD is a simple beast. It runs daemonized, which means that it is loaded into memory and then sits around listening for requests for its services, but otherwise being quite unobtrusive. MPD is a server bound to a port on an IP address. This means that it is capable of supplying its services independently to other computers on the network, although this function is not necessary for standalone use.

**\* This sounds scary, client-server stuff. Is it difficult to set up and use?** No, you only need a few simple lines of text to adjust in the configuration files and a couple of things to type into a terminal, all of which I will show you. The documentation for MPD is not necessarily aimed at beginners. The best of the available documentation is often part of the documentation for a particular distribution such as Arch or Debian, but this can be frustratingly difficult to follow when it deviates from the files and structure of your own distribution. This is the reason for this article, to try to simplify the setup for PCLinuxOS users who want a low resource music playing utility.

The instructions that follow are for a single user system, which makes the set up so much easier, avoiding all those awkward file permission problems. All the configuration files will be in the user's home directory, and will be owned by the user.

As MPD is a server, you also need a client application to request and control the services it supplies. The most basic MPD client is called MPC which, if you hadn't already figured it out, stands for Music Player Client, and it is available from the PCLinuxOS repositories. Although MPC is a command line only utility, it is very powerful and is ideal for setting up the system. MPC is good if you want to control playback of your music from scripts or from short-cut key bindings, such as the multimedia keys found on many keyboards today. I will cover this way of using MPC later.

Most people, though, I am sure would prefer to have a graphical 'front-end' client, rather than a terminal based client, to handle musical things. Fortunately, there are some very nice graphical clients available for us to download and use in the PCLinuxOS repositories. First though, we need to cover the basics of the system, and to know just what MPD and clients, such as MPC, are capable of.

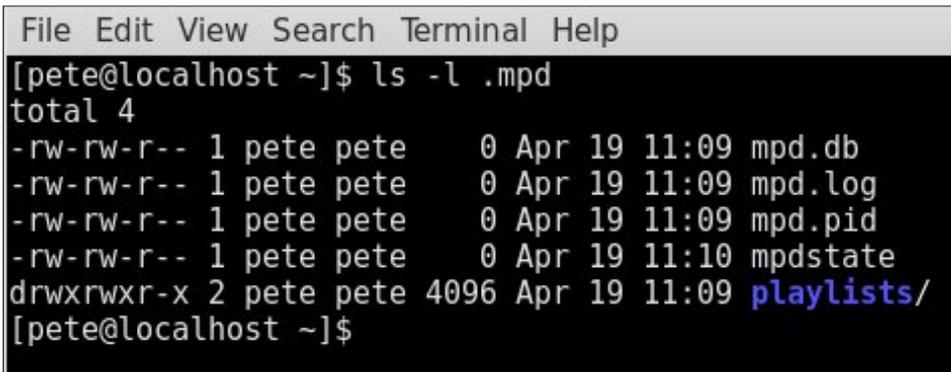
To get started, install MPD (if it is not already installed) and MPC from the repositories, and we'll expand on how to use MPC once it is up and running.

After installing MPD, there will be a global configuration file in the /etc directory named MPD.conf. This is a long file with lots of comments, but way too complicated for the basic system that I have in mind, so I will create a simple configuration file in my home directory. When MPD is started, it looks for a configuration file in the home directory of the user that launches it and, if found, will use that file in preference to the /etc/MPD.conf file. I shall also need to create a few files and directories in my home folder for MPD to use. Having everything in

my home folder means that nothing interferes with the system, and I can edit them without needing root permissions.

The music player daemon needs to know where the music files are stored, as well as where the playlists are. It also needs files to store the music database, its log file, its process id number (so that it may be easily stopped and started), and a file that holds the state of MPD so that it may resume play after stopping. We'll put these files in a hidden directory so that our home directory doesn't get too cluttered. Open a terminal and make sure that you are your normal user (not root) and in your home directory. The following code will then make the required files and directories for us:

```
mkdir -p .MPD/playlists
touch .MPD/{MPD.db,MPD.log,MPD.pid,MPDstate}
```



```
File Edit View Search Terminal Help
[pete@localhost ~]$ ls -l .mpd
total 4
-rw-rw-r-- 1 pete pete  0 Apr 19 11:09 mpd.db
-rw-rw-r-- 1 pete pete  0 Apr 19 11:09 mpd.log
-rw-rw-r-- 1 pete pete  0 Apr 19 11:09 mpd.pid
-rw-rw-r-- 1 pete pete  0 Apr 19 11:10 mpdstate
drwxrwxr-x 2 pete pete 4096 Apr 19 11:09 playlists/
[pete@localhost ~]$
```

Now open your favorite text editor enter the following text, changing the file paths to reflect your own system settings. This will be your own personal MPD configuration file:

```
music_directory      "/home/user/Music"
playlist_directory   "/home/user/.MPD/playlists"
db_file              "/home/user/.MPD/MPD.db"
log_file             "/home/user/.MPD/MPD.log"
pid_file             "/home/user/.MPD/MPD.pid"
state_file           "/home/user/.MPD/MPDstate"
bind_to_address      "localhost"
port                 "6600"
log_level            "default"
audio_output {
type "pulse"
name "pulse audio"
format "44100:16:2"
}
audio_output {
```

```
type "alsa"
name "alsa audio"
format "44100:16:2"
}
```

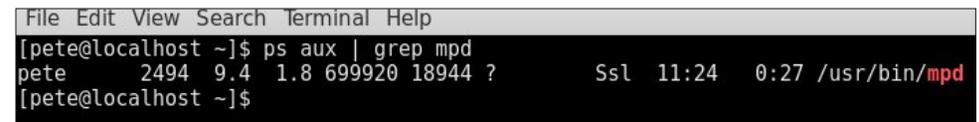
Save the file in your home directory as .MPDconf. Make sure that you include the leading period.

I have two audio output sections here, one for alsa and one for PulseAudio. That's ok. PCLinuxOS can use either, and MPD will choose the one that is currently being used.

Open the PCLinuxOS Control Center and click on System then Manage System Services. Make sure that MPD is checked to run on boot. Now you may have to set MPD to run at login, if it doesn't start automatically. How you do this really depends upon the desktop environment you are using. I'm using Mate, which has a startup applications option in the Mate Control Center. The file that you need to start on boot up is /usr/bin/MPD. If you have trouble discovering exactly how to do this for your desktop, then you might try reading one of the excellent PCLinuxOS magazines special editions: there is one for most of the supported desktop environments. They can be downloaded in pdf format at <http://pclosmag.com/special.html>.

Now reboot the system. This is not strictly necessary, but it will make sure that everything so far is ok, and it should build the database for you. Depending on how many music files you have, this could take a few minutes. You might want to start with a music directory with just a couple of sub-folders containing the music files, and then later update the database to your full music collection once the system is established.

Check that MPD is running:



```
File Edit View Search Terminal Help
[pete@localhost ~]$ ps aux | grep mpd
pete    2494  9.4  1.8 699920 18944 ?        Ssl  11:24   0:27 /usr/bin/mpd
[pete@localhost ~]$
```

If not, then simply type the command MPD to start it up. If MPD was not running but this command starts it and comes back with no errors, then the MPD is not starting at boot and that situation needs to be addressed. Similarly, if you get an error, then you should check the previous instructions carefully.

Now check the files the we made earlier and you should see some changes, after the database has been built. Be patient if you have many, many files.

## Leveraging MPD (Music Player Daemon)

OK! That worked. It's not the best way, but it does prove that MPC and MPD are talking and we now have a playlist to work with. Try the following MPC commands:

MPC stats	To get some database information
MPC play	To play the first file in the playlist
MPC next	To play the next file
MPC play 6	To play the sixth file in the playlist
MPC pause	To pause playing

Type MPC help in a terminal to see a long list of the commands that MPC can accept.

```
File Edit View Search Terminal Help
[pete@localhost ~]$ ls -l .mpd
total 1228
-rw-rw-r-- 1 pete pete 905128 Apr 19 11:26 mpd.db
-rw-rw-r-- 1 pete pete 341411 Apr 19 11:26 mpd.log
-rw-rw-r-- 1 pete pete 5 Apr 19 11:24 mpd.pid
-rw-rw-r-- 1 pete pete 0 Apr 19 11:10 mpdstate
drwxrwxr-x 2 pete pete 4096 Apr 19 11:09 playlists/
[pete@localhost ~]$
```

The database file should now be quite large. If not, try executing MPC update and wait a minute or so. This uses the client utility to tell MPD to update its database.

We are now ready to test the set up. Bear with me on this - MPC is not the easiest client to use manually, but for testing purposes, it is fine. To play music, we need a list of files to play. A playlist is simply a text file containing a list of the filenames of music files and MPD will locate these files, if they exist, in its database. The playlist should be saved into the playlist directory we created, and should have an extension of .m3u.

As we want to make sure that everything is set up correctly, we will use MPC's search feature to create a playlist for us. The MPC client searches the metadata found in music files, and can search on fields such as artist, album, genre etc. Choose an artist that you know is in your collection. We'll do the search and redirect the results straight into a playlist. The search command finds the files, the load command loads a playlist for use and the playlist command displays the list of songs available in the playlist.

```
File Edit View Search Terminal Help
[pete@localhost ~]$ mpc search artist sade > ~/.mpd/playlists/sade.m3u
[pete@localhost ~]$ mpc load sade
loading: sade
[pete@localhost ~]$ mpc playlist
Sade - Like a Tattoo
Sade - Cherish The Day
Sade - I Miss You (remix)
Sade - No ordinary love
Sade - Kissing You
Sade - Smooth Operator (extended smooth jazz version)
Sade - Paradise
Sade - By Your Side
Sade - Flow
Sade - King Of Sorrow
Sade - All About Our Love
Sade - Slave Song
Sade - The Sweetest Gift
Sade - Every Word
Sade - Immigrant
Sade - Lovers Rock
Sade - It's Only Love That Gets You Through
[pete@localhost ~]$
```

```
File Edit View Search Terminal Help
[pete@localhost ~]$ mpc stats
Artists: 771
Albums: 247
Songs: 4129

Play Time: 0 days, 0:00:00
Uptime: 0 days, 0:42:35
DB Updated: Sun Apr 19 12:21:08 2015
DB Play Time: 10 days, 16:41:47
[pete@localhost ~]$ mpc play
Sade - Like a Tattoo
[playing] #1/17 0:00/3:31 (0%)
volume: n/a repeat: off random: off single: off consume: off
[pete@localhost ~]$ mpc next
Sade - Cherish The Day
[playing] #2/17 0:00/0:07 (0%)
volume:100% repeat: off random: off single: off consume: off
[pete@localhost ~]$ mpc play 6
Sade - Smooth Operator (extended smooth jazz version)
[playing] #6/17 0:00/8:51 (0%)
volume:100% repeat: off random: off single: off consume: off
[pete@localhost ~]$ mpc pause
Sade - Smooth Operator (extended smooth jazz version)
[paused] #6/17 0:35/8:51 (6%)
volume:100% repeat: off random: off single: off consume: off
[pete@localhost ~]$
```

If you got similar results, you're good to go and we can look at an easier to use, graphical client. Then perhaps, we will see what this thing can do.

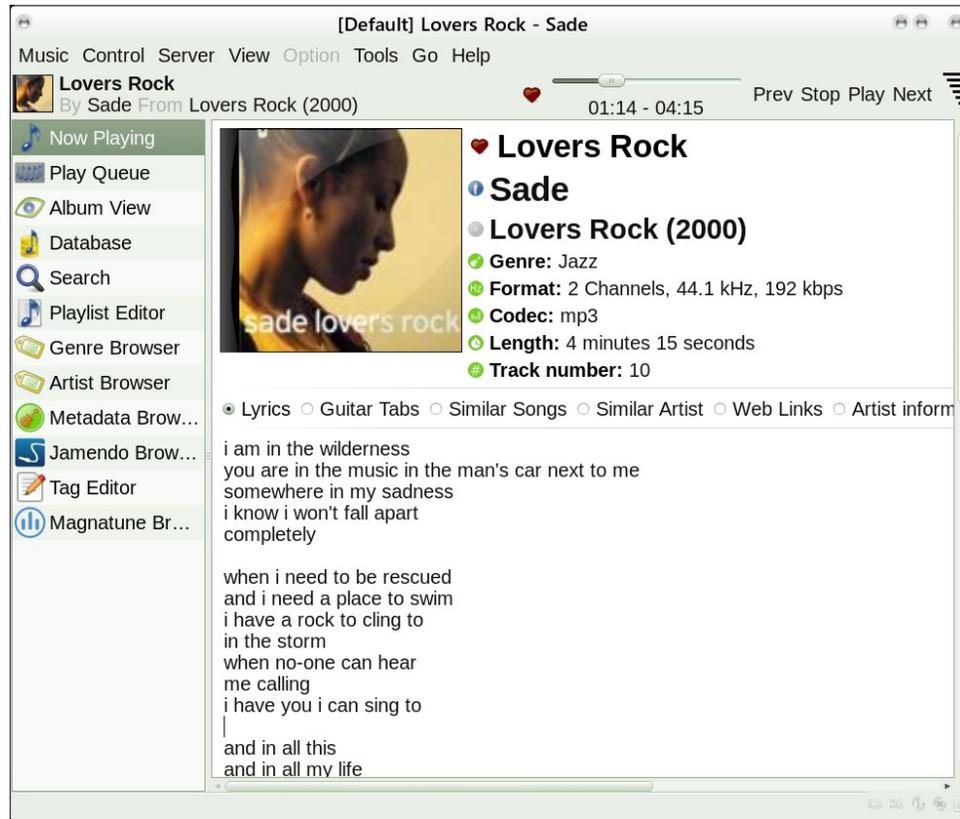
The music player daemon, MPD, is capable of satisfying the needs of underachieving souls like myself through and up to audiophiles so demanding that they like to tune their own pianos, from 'graphical interface only' users to



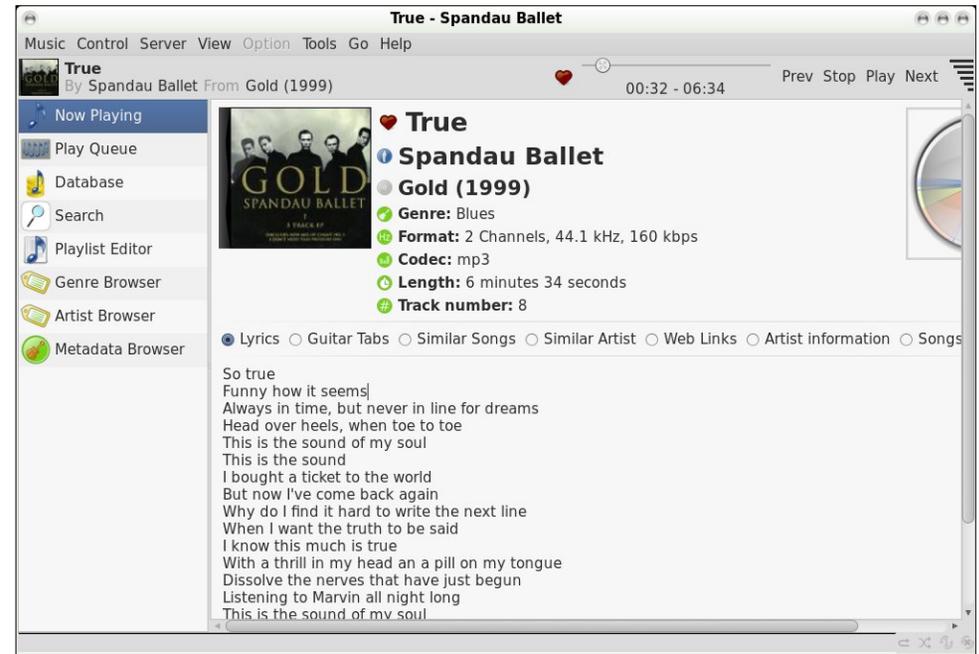
bearded geeks who live in a monochrome world of tty terminals and kernel optimisation. Me, I prefer simple, but with a little cream.

Judging from the number of client applications that have been written to interface with MPD, it just must be good.

One of the very best graphical clients for me is Gnome Music Player Client or gMPC, and luckily for us, it's available in the PCLinuxOS repositories. On its first run, you are presented with a set up wizard dialog, but you can safely just accept the defaults and then explore things later.



Everything that you might need is in there including graphics, lyrics and a playlist editor. Create a new playlist, jump to the database and find the files you want to add, select copy, jump back to the playlist editor and right click to paste them in. It's really quite intuitive.



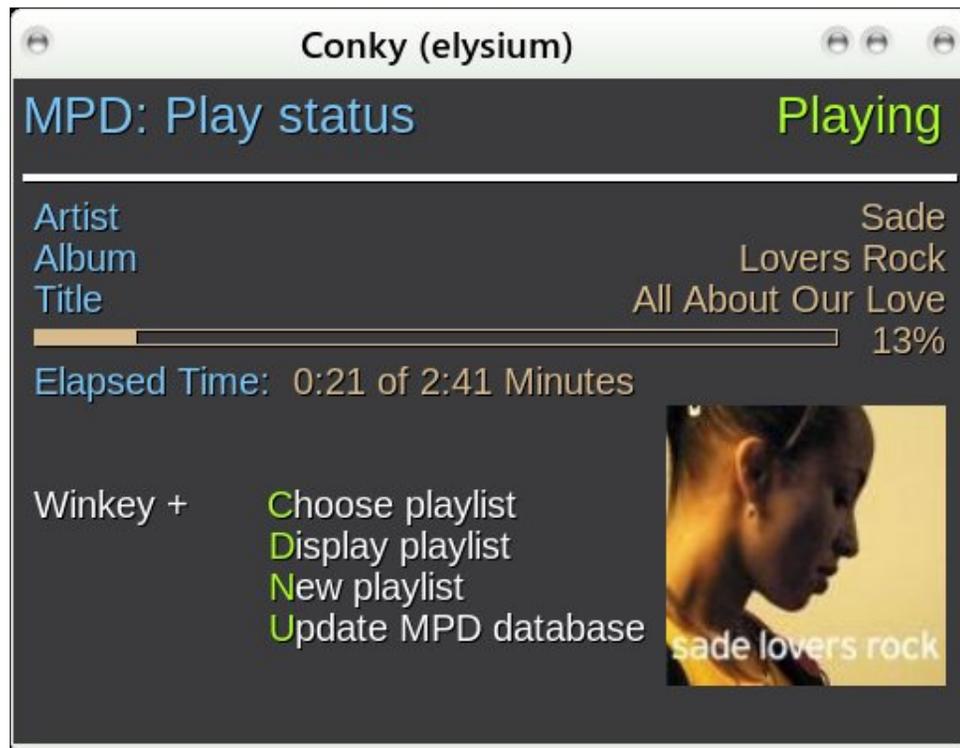
I really don't need to tell you any more. Just try it, explore it and enjoy it.

There are other client applications available from the PCLinuxOS repositories. PyMPD is a nice, minimalist player written in python. Sonata threw up permission problems for me, so I abandoned it. ncMPC is a curses-based client designed to be run in a terminal which, in these days of graphical environment domination, seemed rather pointless. If you need a terminal based client then ncMPC is probably the one to go with, as it is very comprehensive. You pays your money...

My own preference is to utilise conky, the system monitor, which has a whole slew of MPD connected features. This, in conjunction with a few shell helper scripts and our old friend MPC, does exactly what I want. No more than I want or need, so it is perfectly optimised to my own system.



My music files are stored as MP3 files, each album in its own folder with multiple volumes in subfolders, and each folder or subfolder containing album art in the form of a single file named 'folder.png.' The size of the graphics file is not particularly critical, as it is scaled by conky to 150x150 pixels to suit my display. If the image is missing, then a default image is shown.



It's difficult to say just how efficient with resource usage this system actually performs. This set up with MPD, & MPC seems to be using about 13 MB of memory, and conky another 17MB. Allowing a little more for caching of the read ahead buffers, let's say 35MB tops. CPU usage on my dual core laptop processor increases by only around 1% when activated and playing music, which is pretty good. It shouldn't impact other applications, even on very minimal systems.

As MPD utilises playlists, I decided to add a little playlist creation script, imaginatively named create-playlist.sh. It will allow you to visit folders in your music collection and automatically create a playlist, overwriting any existing playlist. It then opens the playlist in a text editor for you to organise the tracks. The final playlist is copied to both the music directory and to the MPD playlist directory so that it is immediately available for selection. There are better playlist editors out there but this one does the job for me and, having the script code, I can alter it as time demands.

If you want the lyrics, then the final script should help. If the lyrics are not available on that site, you'll get the message "Sorry, We don't have lyrics for this

song yet,” but most popular songs seem to be there. The output from the script is not perfect, but I decided to accept it as it is rather than spend too much time on refining it. You will need to have wget and zenity installed from the PCLinuxOS repositories to use the scripts.



I've included my code below for anyone who wants a heads up on controlling MPC with scripts, and on showing the activity in conky. I have highlighted the parts that will need to be changed to work on a different system.

Start conky with `conky -c ~/.conkyrc_MPD`.

Script 1 is called automatically by the conky resource file.

Scripts 2-4 are bound to hotkeys as shown in the conky display.

Script 5 I have bound to the multimedia start/stop music key on my laptop.

Script 6 Fetches the lyrics of the current song.

Conky color references can be found here:  
<http://www.graphviz.org/doc/info/colors.html#x11>.

### Conky resource file .conkyrc\_MPD

```
double_buffer yes
own_window yes
own_window_hints below,skip_taskbar
own_window_transparent no
update_interval 1.0
minimum_size 500 50
use_xft yes
xftfont Sans:size=10
own_window_colour 3b3b3e
color1 ghostwhite
color2 skyblue
color3 chartreuse
color4 navajowhite3

TEXT
${font Liberation Sans:size=16}${color2}${if_running MPD}\
MPD: Play status${alignr}${color3}${MPD_status}
${color}${hr 4}${font Liberation Sans:size=12}
${if_MPD_playing} ${color2}Artist${alignr}${color4}${MPD_artist}
${color2}Album${alignr}${color4}${MPD_album}
${color2}Title${alignr}${color4}${MPD_title}
${MPD_bar 8,430} ${alignr}${MPD_percent}%
${color2}Elapsed Time: ${color4}\
${MPD_elapsed}${color1} of ${color4}${MPD_length} ${color}Minutes
${color1}Winkey + ${color3} C${color1}hoose playlist
${color3} D${color1}isplay playlist
${color3} N${color1}ew playlist
${color3} U${color1}pdate MPD database${endif}${endif}
${execi 1.0 /home/user/bin/MPC-art.sh 2>/dev/null}
${image /tmp/MPD.png -s 150x150 -p 345,170 -f 1.0}
```

## 1. Script MPD-art.sh

```
#!/bin/bash
trackpath=$(MPC current --format %file%)
trackdir=$(dirname $trackpath)
filepic="/home/user/data/Music/"$trackdir/folder.png
# if no graphic then supply a default image
defaultpic="/home/user/.MPD/folder.png"
if [[ -f $filepic ]]
then
    /bin/cp -f $filepic /tmp/MPD.png
else
    /bin/cp -f $defaultpic /tmp/MPD.png
fi
exit 0
```

## 2. Script create-playlist.sh

```
#!/bin/bash
# most of the work is done in this function
makelist() {
# navigate to the required directory
pldir=$(zenity --file-selection --directory --
filename=/home/user/data/Music/)
cd $pldir
# generate a name for the playlist
# Remove the path to the music directory
tmpname=$(echo $pldir | sed -n 's/\home\/user\/data\/Music\/\//p')
# replace any subdirectory slashes with dashes
pname=$(echo $tmpname | sed 's/\//-/g')
# clear any previous playlist of that name
> $pname.m3u
# generate the playlist
for mp3file in *.mp3
do
    echo $mp3file >> $pname.m3u
done
# open playlist for editing
pluma $pname.m3u
# copy playlist to MPD playlist directory
cp -f $pname.m3u /home/user/.MPD/playlists
} # end of function makelist
# script starts here
makelist
# go round or quit
zenity --question --text="Make more playlists?"
while [ $? = 0 ]
do
    makelist
```

```
zenity --question --text="Make more playlists?"
done
exit 0
```

## 3. Script new-playlist.sh

```
#!/bin/bash
# get a list of playlist names
plists=$(ls /home/user/.MPD/playlists | sed 's/.m3u//')
# display a dialog to choose a playlist
pl=$(zenity --list --title="Playlists" --column="Available
Playlists" --width=350 --height=350 $plists)
if [ -z $pl ] # exit on cancel
then
    MPC play # no new playlist selected so resume
    exit 0
else
    MPC clear # clear current playlist
    MPC load $pl # load new playlist
    sleep 2 # allow the new playlist to complete loading
    MPC play # start playing new playlist
fi
exit 0
```

## 4. Script show-playlist.sh

```
#!/bin/bash
# get a list of playlist tracks
tracks=$(MPC playlist)
# display a dialog to show the tracks
IFS=$'\n' tl=$(zenity --list --text="Current MPD Playlist" --
title="Available Tracks" --hide-header --column="Available tracks"
--width=600 --height=450 $tracks) 2>/dev/null
exit 0
```



5. Script conky-MPC-toggle.sh

```
#!/bin/bash
MPDPID=$(ps aux | grep "[c]onkyrc_MPD" | awk '{print $2}')
if [[ -z $MPDPID ]]
then
    MPC toggle
    /usr/bin/conky -c /home/user/.conkyrc_MPD
else
    MPC stop
    kill $MPDPID
fi
```

6. Script get-lyrics.sh

```
#!/bin/bash
# get the artist and song title of the current track
artist=$(MPC -f %artist% | head -n 1)
title=$(MPC -f %title% | head -n 1)

# clear any previous lyrics
song=""

# Download the lyrics
song=$(wget -qO-
"http://makeitpersonal.co/lyrics?artist=$artist&title=$title")

# Split the lines where there is a space followed by an uppercase
letter
# Then pipe the output to zenity
echo $song | sed 's/ [[:upper:]]/\n&/g' \
| zenity --title="$title" --text-info --width=400 --height=640
```



## The PCLinuxOS Magazine Special Editions!

**Get Your Free Copies Today!**



# PCLinuxOS Family Member Spotlight: wayne1932

## As Told To Smileeb

For this month we have wayne1932



Hey, when you really reach the bottom of the barrel, you might consider me for the magazine article. Short info: I've been hanging around PCLOS for quite a while, started with Mandrake, and found out that Tex made better packages than the Mandrake guys. When he started his own distro, I moved over. I notice that I've got three years on you. My forum name includes the year I was born. :)

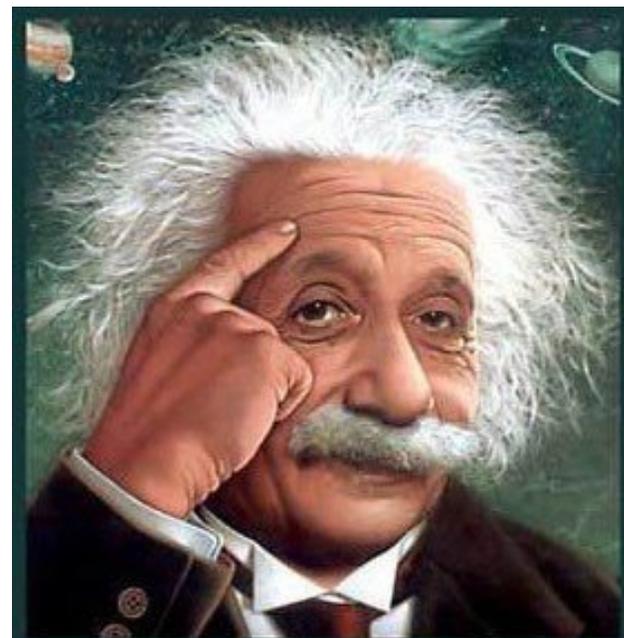
I'm 82 years old, born late in 1932. (Genuine Okie from Muskogee, for the country western music fans here). I called Muskogee home for the first twenty seven years of my life. After high school, was unable to go to college (finances), so took a job with the railroad and worked up to a telegraphers position (clicketyclack) Then I was drafted into the army for two years for the Korean war. Furthered my interest in electronics by being a radar operator. Army provided me with GI bill so off I went to college for my Bachelor of Science in Electrical Engineering.

After graduation worked 29 years in the telecommunication field as R&D engineer inventing equipment for the long distance network. Unfortunately, I specialized in analog equipment and was obsoleted too young in a digital world. I obtained my masters degree during that work career. I was involuntarily retired at 56 YO, so kept working as a consultant in telecommunications. Finally decided in 2000 that 40 years was long enough for any one in one field, so hung up my hat and stayed home in retirement.

I have spent the years since 1960 living in the Dallas Texas area, met my wife here and raised two great independent kids. Unfortunately independent means I have no grandkids. Dallas is a great place to live, getting too overcrowded now, but still has all the amenities for a good interesting life.

My wife of 52 years and I now live a quiet life being involved with our church. After our kids were old enough to fend for themselves, she went to work in a variety of places as office administrator. Her last position was with Nokia, so I got an education in cell phones. We loved traveling and spent many years using a camper/trailer/ RV seeing the great sights US of A. I have been in every state except Delaware, may have to make a special trip just to say I've been there. I have spent some time researching my genealogy as far back as I can.

My computer experience began the day in the early 80's they put an IBM machine on my desk with TWO floppy drives and 64K of memory, with a dot matrix printer. I got proficient in WordStar, and Visicalc. Worked my way up from DOS 1.1 thru Windows 98SE, and decided I didn't like Microsoft's business practices. I started Linux using Mandrake, and found that Tex made better and more interesting packages. When he started his own distro I went with him and been here ever since. I keep my toe in the water and have a special part of my harddrive dedicated for my experiments with other distros, but nothing has tempted me to jump ship yet. I have always loved KDE for its ability to be configurable. My wife and I have both a Laptop and Desktop machine networked thru Samba. Unfortunately I haven't been able to get her to switch to Linux. She claims that she doesn't want to learn something new. HaHaHa..... Win7 was a worse experience than the learning curve for PCLOS would have been, and to top that off, it meant I had to learn it too.



**It's easier than  $E=mc^2$**   
**It's elemental**  
**It's light years ahead**  
**It's a wise choice**  
**It's Radically Simple**  
**It's ...**



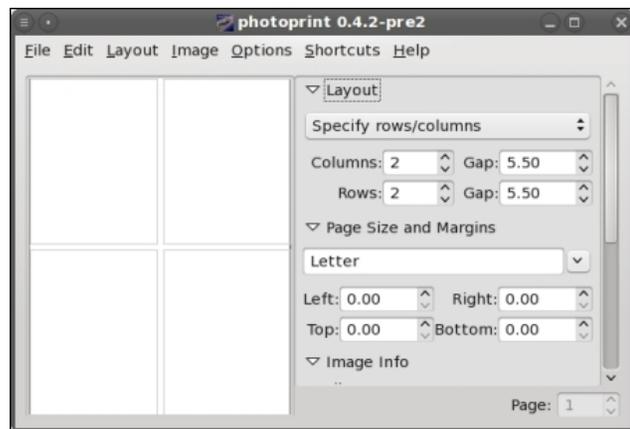
# PhotoPrint: Application For Fast Photo Layout And Printing

by Khadis

Printing photos is not always an easy thing to do. I mean, we cannot always just right click on a photo then press "print." Sometimes we need to adjust the layout, too. Although Linux provides sets of applications related to photo management and photo printing, and you might have one that you really love, I will show you another option (if you have never tried it yet). The application is PhotoPrint, and it is a friendly application, even for a newbie.

The basic operation of PhotoPrint is printing any photo as you need. You can choose to set the photo(s) to fit the paper size, set one piece of paper to load many photos at once, or even to set the photo size by the need.

To print several photos at once, you can first choose the layout method. There are two methods that you can use: **specify rows/columns** or **specify image dimensions**. Example: when you want to fill the paper with 4 photos, you can set the parameter as follow (units are in millimeters):



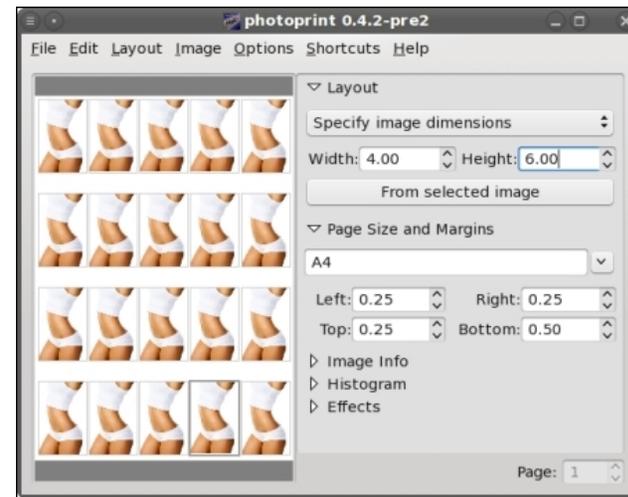
We can set the number of rows and columns to place the photos and set the spaces or gaps among them. We can even set the margins and paper size. The default unit is millimeter, but you can change it into centimeter or inches from menu **Options – Units**.

We can also set the size of photo that we want to print. Example: we want to print a 4 x 6 cm photo, we can set the parameter as follow (units are in centimeters):



To add photos into the PhotoPrint, we can do it easily by pressing **Ctrl + I** and then access the directory where we saved the photos. Once the photo appears, we can select other photos to put them into the remaining rows or columns. If we want to fill the whole paper with several of the same photo, just click on the first photo, then click **Image – Duplicate to Fill Page**. It's good for mass photo printing (top, right).

There are 4 layout options we can use: auto, poster, carousel, and manual. If you are planning to print a large photo onto a small size paper, you can choose poster layout mode. The photo will be printed in sections onto several pieces of paper.



PhotoPrint also provides a few effects that can be applied. They are Desaturate, Warm/Cool, and Sharpen. They are not for professional use, but maybe if you want to make your photos little bit "different," then you can use one of those effects.



# YAT: Yet Another Testimonial

by Odin

After reading so many testimonials (and writing one myself, I think, a long time ago), I have to say it again: PCLinuxOS is the best, most reliable distro ever, at least for me.

I come from Windows, like so many others, and I've always been dissatisfied with the bloat, the malware, and all the other, too numerous to mention, drawbacks that plague Windows.

Tried too many distros: some don't work at all, some are too obscure to operate, some are fine on Live DVDs but stop working when installed, some don't even see my wifi, some don't partition correctly, some don't have sound, etc. And it's not a machine-specific thing. I've changed gears many times across the years, and what I have consistently found was that in 100% of the cases, PCLinuxOS installs easily and works, out of the box, perfectly, and for a very, very long time.

It worked, and works, so well, that it was always an excuse for me to try other distros because, in my mind, if PCLinuxOS does its thing so well, the others must equally do so. Very wrong! I am amazed at how poorly other distros compare, and at how some simple things, which I take for granted now in 2015, can go so wrong.

So this is yet another testimonial to the greatness of PCLinuxOS, and how this distro, ranked now 13th on Distrowatch, deserves, in my opinion, to be no less than number 1.



I have been so happy with PCLinuxOS that I have finally (and I hope for good) removed Windows (this time Windows 8.1) from my computer completely. It is true that I miss gaming, and I hope more work is done in the future to have this pending issue solved for good.

Kudos to all, and to the friendly and helpful community on this forum!



Visit.  
Contribute.  
Build.

The PCLinuxOS  
Knowledge Base

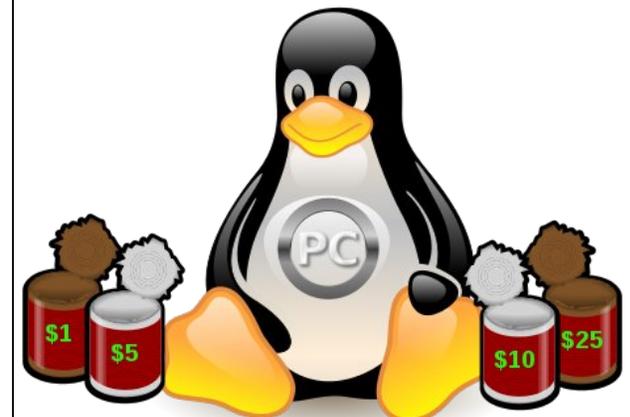
*It Belongs To YOU!*

## Donate To PCLinuxOS

*Community Supported.  
No Billionaires/Millionaires.  
No Corporate Backing Or Funding.*

Click [here](#) to make a one-time donation through Google Checkout.

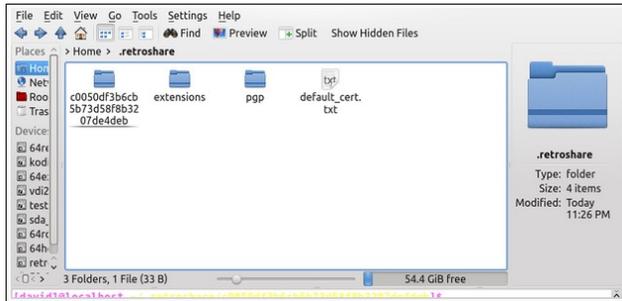
Or, click one of the amounts down below to make a monthly, recurring donation.



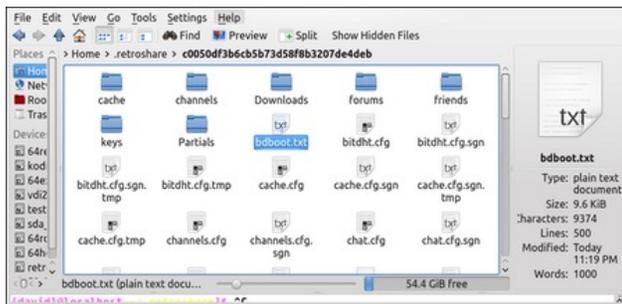


And then you should be able to see the **/retroshare** folder in the Dolphin file manager.

Open the **.retroshare** folder.



Open the folder with all the numbers in there.

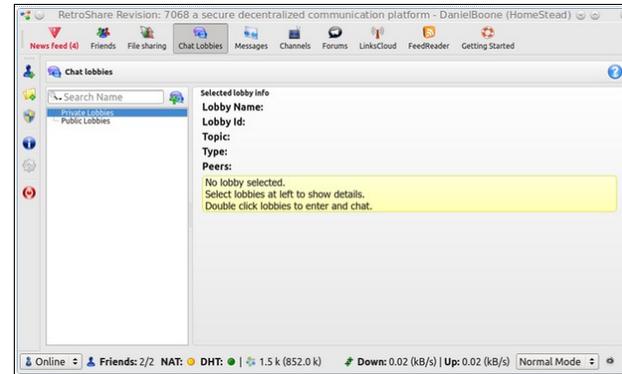


There will probably not be a **bdboot.txt** file inside your folder yet. You have to put that inside there manually, and on each new install or upgrade you will have to reinstall the **bdboot.txt** file. I have it in my Dropbox folder, from which you can download it, if needed. Download the file from my Dropbox and save it to the `/home/<user-name>/retroshare/<your user number>/` directory.

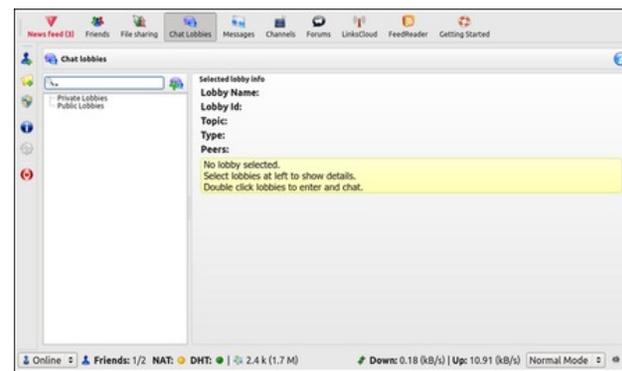
Here is the URL:

<https://www.dropbox.com/s/xx9s5jevawm8ebx/bdboot.txt?dl=0>

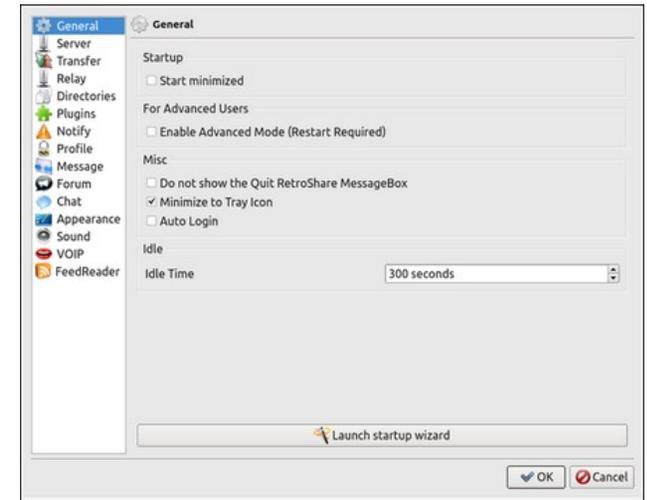
When you do get to the GUI, it may take up to 30 minutes for the DHT light to turn to green, but after we get the settings right, your DHT Light will go green between 10 to 20 minutes. If you have read the wiki, it will tell you it can take up to 30 minutes. Usually, though, mine will turn green within a few minutes. As for the NAT Light, mine stays yellow most of the time, but it does turn green sometimes, but will not stay that way.



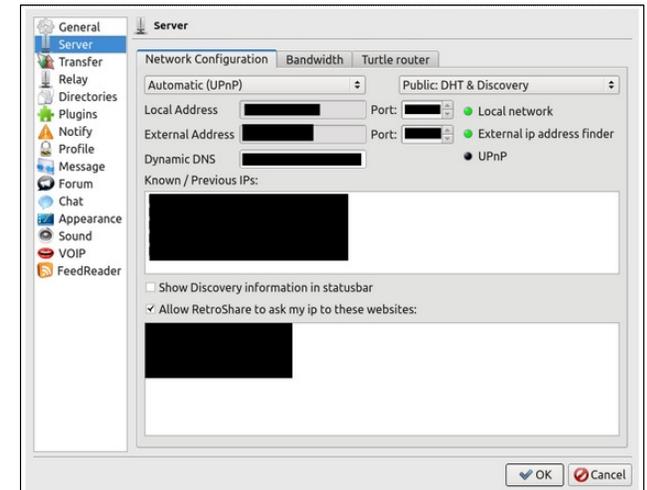
Now we have to set up a few things, so let's go through that so we're into the main GUI now.



The 5th icon on the left side is a little gray gear. This is your options, so click on that and it will bring you to this screen (top, right).



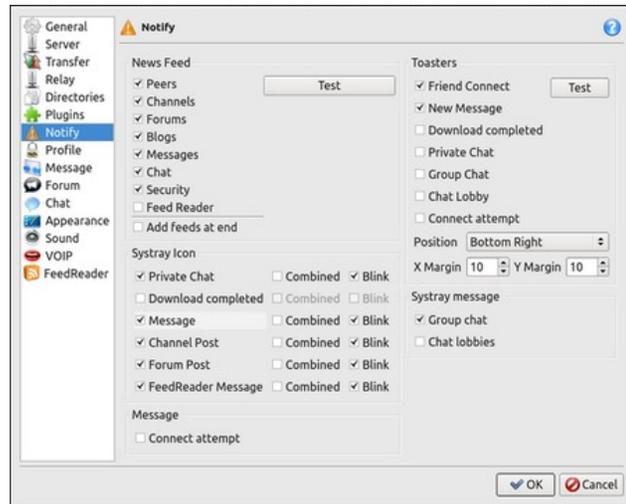
Then click the server tab ...



The blocks you see blacked out above will not be in yours, if you set it up right so far. Local and external addresses should auto configure, along with the port settings. You can set the top dropdown to "firewalled" if you wish, but what I have read is that most of the time the way mine is set will work, also. Use your discretion. As for "Public DHT

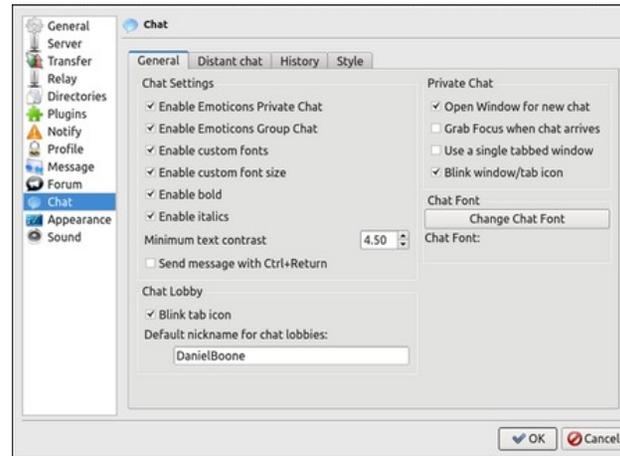
& Discovery,” you may have to switch to "Inverted DHT." If you have read the info prior to the installation of RetroShare, you can find out about that. The only thing you will really have to do manually is add your "Dynamic DNS" yourself.

In the notify option, you can tick what you want to tick. You will have some that are ticked and some not. I did set mine to "blink" when I get a new message, but that was my decision. You do not have to do it, if you wish. It just makes it a little easier to see when someone has sent you a message. Of course, you need to have the other user's encryption code. In time, we will pass encryption codes to each other as we gain more users and exchange keys between each other, and not on the Open Forum.

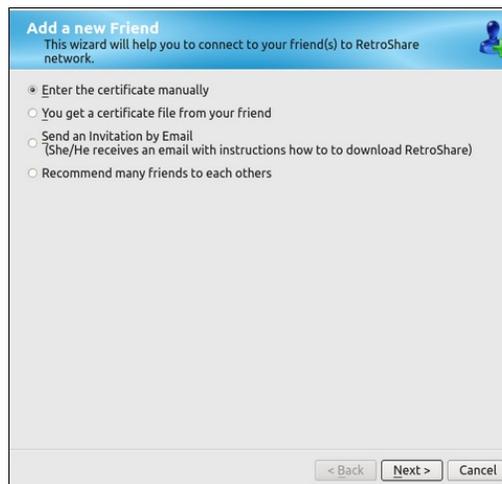


Something you should also check, once you have RetroShare up and running, is if you can send a message and use the enter key to post the message. Let's take a look at the snapshot at center top.

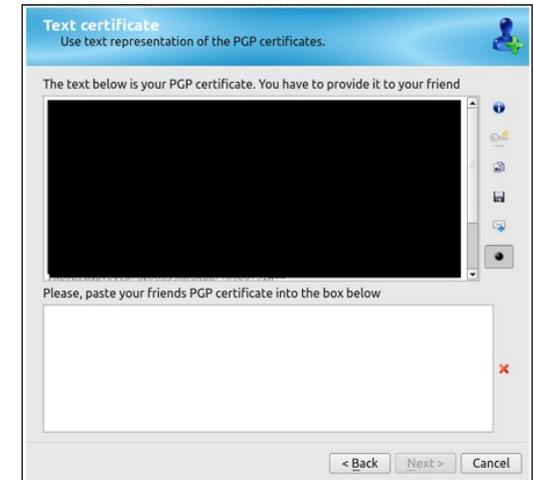
Make sure the box beside "Send message with Ctrl+Return" is **unticked**.



We are going to add a new user for a contact. Select the first icon on the left side. It will bring you to this window:



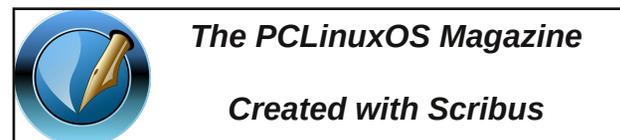
Click on "Next" and this is where you will enter your friend's key code (top, right).



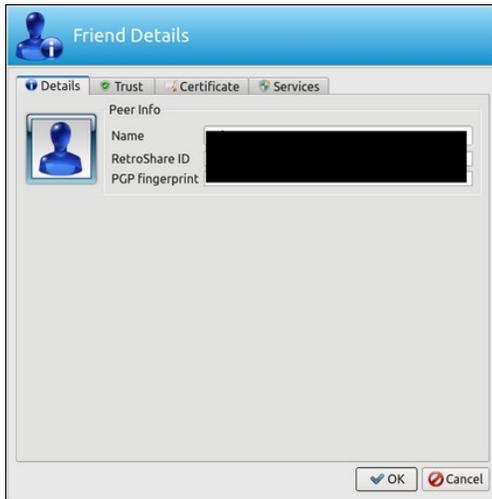
You will notice I have blackened out my encryption, but yours will automatically be in the top "blacked out" box. In the second block, this is where you will place your friend's encryption code. Click the "Next" button, which will bring you to this window. If their encryption code is good, the "X" will change to green. Click "Next" again.



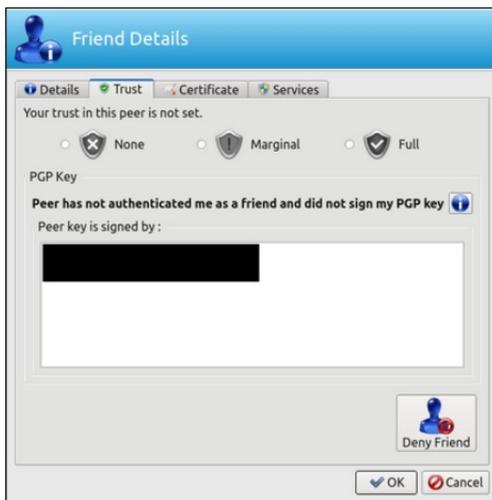
Again, the blocks I have blackened out are to protect the identity of my friend. If yours looks like this, then we are done. Click the "Finish" button.



We now have to set permissions for our friend. Click on the top highlighted tab. You will notice that I have blackened out my friend's information again, but right click on your friend's name, and it will open a drop down selection box. Select "Friend Details," which will bring you to this window:



Click on the "Trust Tab."



I granted "Full" trust to my friend. Down the left side of the window, across from the Deny Friend, will be

one list for you to sign your friends key. Do so, then click OK.

Now, go back to the main window, and on the left panel, click on the folder icon. We have to set the permissions for our share files, and that will bring you to this:



Yours should look like the top entry, since that will be the only one you have listed at this point. Now, notice how the folders appear. This is not how you want them to appear. You want them to look exactly like the other two I am showing. So, click on each folder and make one and two a folder, as I have done, and set the third one as a color.

As for the other two that you see in the image, you can simply add them to yours with the "Add" button. You do not have to do it at this time. That part is up to you, so click on the "Apply and Close" button.

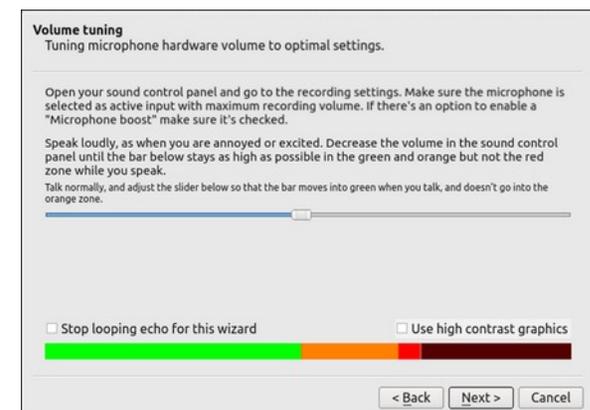
I think I have covered all I can to get you up and running. As far as adding friends, you will have to PM me. Ask me for my RetroShare keys, and I will add you and send you my personal encryption code so you can add mine. As we go along with this, a lot of us will probably share encryption keys with each other.

Presently, I know three of us from the forum are using RetroShare, and trying to work out bugs. I have been able to make RetroShare pick my headset and mic. Just to show you how, let's go back to the main window. Click on the VOIP Button,

and that will bring you to this, but make sure you have your mic and headset plugged in before you start. The only way I found to make it work is to disable PulseAudio in the PCLinuxOS Control Center (PCC), and use straight ALSA. Then, and only then, can I get RetroShare to discover my headset and mic. One of my other friends cannot get RetroShare to find his mic, so things may vary on this issue. I have got an old headset I bought from Radio Shack 12 to 15 yrs ago, and it does find mine. I also want to mention now that while I can get my VOIP option to work, and I can hear mine, the "SIP protocol" is only half way finished, so it may not work at all. That will be up to the developers to do their work.



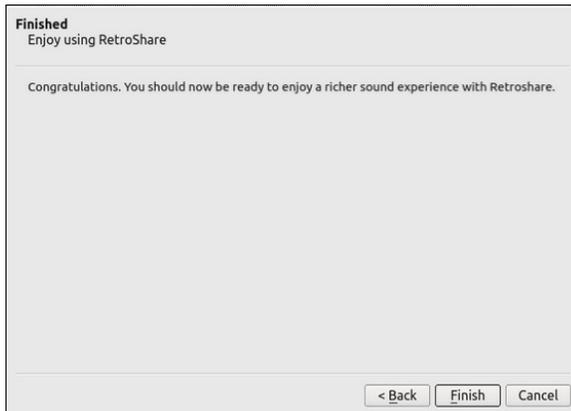
Click Next, and that will bring you to the next window.



So, that will bring you to this window:



So if yours looks like this, then click the “Next” button again and that will bring you here:



If yours looks like this then select the “Finish” button, and you’re good to go.

I hope all goes well with your installation of RetroShare. I am by no means a expert on this, but me and some of my buddies may be able to help you along the way. Best of Luck.

My best to all. This is DanielBoone, or David1958 as some know me. I got cabin fever over the winter, and decided to change my forum name.

Don't forget ... what I have written are only suggestions, so please take them as such, and make your own decisions about the changes mentioned.



## Screenshot Showcase



*Posted by luikki, May 1, 2015, running KDE.*

# Game Zone: Neon Struct

by daiashi



## About The Game

You are Jillian Cleary, an ex-spy on the run, framed for treason and hunted by your former agency. You must move in secret, survive off the grid, and find a way to clear your name. Old friends and new allies will lend their support, but with the eyes of the world upon you, who can you trust?

## Stealth

Keep in the shadows, move in silence, and stay out of sight. Escape from tight situations by scrambling your enemies' vision. Enhance your performance with black market stims.

## World

Explore a concrete-and-neon brutalist vision of the modern surveillance state. Undertake eight missions spanning two continents. Meet a diverse range of interesting characters.

## Music

Featuring an 18-song soundtrack by electronic rock band The Home Conversion.

Simple but interesting game. Slight Deus Ex feel. A little suspense and action all rolled into one. It is very slow paced due to interaction with others, including your contacts. I found it rather enjoyable and hope you will, too.

## System requirements:

Fully updated PCLinuxOS and Steam

## Hardware:

### Minimum:

OS: PCLinuxOS  
Processor: Intel Core 2 Duo or equivalent  
Memory: 512 MB RAM  
Graphics: OpenGL 2.1-compliant device with 512MB memory  
Hard Drive: 200 MB available space

### Recommended:

OS: PCLinuxOS  
Processor: Intel Core i5 or equivalent  
Memory: 512 MB RAM  
Graphics: Dedicated graphics card, GeForce 9/Radeon HD 3000 series or better  
Hard Drive: 200 MB available space

## About The Company

## Developer

David Pittman makes video games. In 2013, he co-founded Minor Key Games and released Eldritch. He was formerly a programmer at 2K Marin, where he wrote AI code on BioShock 2, and led the AI team on The Bureau: XCOM Declassified. He holds a Master of Interactive Technology degree from The Guildhall at SMU.

## Publisher

Minor Key Games was formed in 2013 by game industry veterans and twin brothers David Pittman and J. Kyle Pittman. NEON STRUCT is their fourth

published title, following Eldritch, You Have to Win the Game, and Super Win the Game. Minor Key Games is based out of Coppell, TX and Novato, CA.

## Some Gameplay Screenshots





### Getting It To Run

Install Steam (if you don't have it installed already), then start it. You will need to create a new account, if you do not already have one. Once you have Steam up and running, go to the store tab. Click on the Linux tab and search for Neon Struct. Click on and download the demo. If you have updated your system, including graphics drivers, you should be good to go. You can also get it here: <http://store.steampowered.com/app/368320/>.

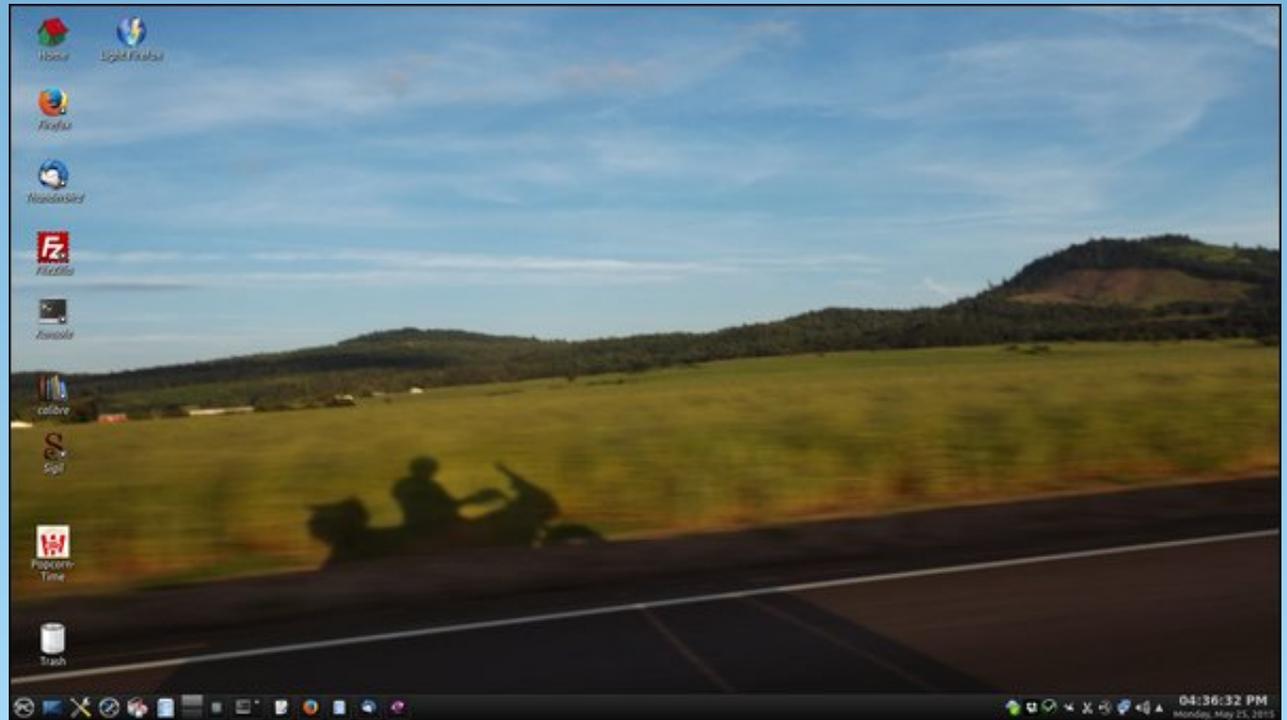


**The PCLinuxOS  
Magazine**

**Created with  
Scribus**



## Screenshot Showcase



*Posted by Mr Cranky Pants-YouCanToo, May 25, 2015, running KDE.*



# Monitor Gamma Correction Made Simple

by critter

Following the recent death of my desktop computer, I was left with a nice, large spare monitor. It seemed a shame to have it just lying around doing nothing, so I decided to connect it up to my laptop and have a dual display.

I was very pleased with the result, working on the laptop, but having reference material available in an easy to read format right next to it (and I had a large screen to play videos on), was so much better than my restrictive laptop screen.

To complete my new set up, I found a nice wide screen format background that I could span across both screens. Nice!

That was when I noticed it. The colors on each screen were completely different. Now where my wife, who graduated from art college and is a professional textile designer, would have noticed this immediately, I didn't see anything wrong until I had the two halves of the picture in front of me.

Now that I had seen this difference, and it annoyed the hell out of me.

I fiddled around with all the adjustments I could find, but either one was too watery or the other like a Martian twilight. I turned to the internet for help. After an hour or so of searching I determined that my problem was something called gamma correction. I don't pretend to understand this, but it seems to be a bit like contrast. Fortunately, there is a utility available for Linux users called xgamma to control this, and it is in the PCLinuxOS repositories. Now it's going to be easy.

No, it isn't. This utility allows me to change a number from one to something else for either all of this gamma stuff or for each red, green or blue component part of it. After about fifty or so attempts at 'correcting' my gammas, I had to reset everything back to default values to get some sort of sanity back to my displays.

I may not know what this gamma thing is all about, but I would know if it looked right to me and that is all that matters – right?! What I need is some sort of instant feedback from xgamma so that I can see the effect as it is applied and gradually, slowly reach some sort of compromise to perfection. Something with slider controls that would allow me to inch slowly towards my goal.



Fortunately I had the basics of such a tool left over from a series of articles I did for the magazine last year about using gtkdialog to add graphical interfaces to scripts. The example I had in mind was used to rotate and resize an image. I decided that I could reuse most of the code, rip out the parts I didn't need such as rotation and the images, redefine the scaling slider to adjust the red gamma and add two more similar sliders for the green and blue gammas.

To use the xgamma utility you supply new values for each of the red, blue and green components that you want to adjust. For example:

```
xgamma -rgamma 1.15 bgamma 0.97
```

From this you can see how hit and miss this technique is. By typing a value of 0 for the red gamma I was told bluntly that acceptable values were between 0.100 and 10 so these should be the limits for the sliders but during my previous attempts I found that anything above 2 was almost offensive therefore 0.1 to 2 would be used. The default value is 1 and so that is to where the sliders should default.



I needed to add a timer so that the values would be updated as I made the adjustments and this would give me the real time effect I was looking for. I had used a timer in the final gtkdialog article that I had written for the magazine and so this too was easy to implement. I found that an update interval of 200 milliseconds gave quite

smooth results. As most of this 'programming' was simply copy, paste and delete text editing of existing code, it took me a little over half an hour to get a usable utility, less than the time I had spent blundering about previously.

It worked admirably. I added an entry to my start up applications so that the adjustments are made automatically on startup.

```
xgamma -q -rgamma 1.28 -ggamma 1.04 -bgamma 0.94
```

My twin monitor set up is not perfect but neither is it annoying or embarrassing and I have another little utility in my toolbox.

## The code

Note that the lines in blue should be three lines, each beginning with <action> and ending with </action>. Splitting them will result in an error.

```
#!/bin/bash
```

```
MY_DIALOG='<window title="Gamma correction utility"
resizable="false">
```

```
<vbox>
  <vbox>
    <frame>
      <vbox>
        <frame Red>
          <hscale width-request="800" height-request="80"
            range-min="0.1" range-max="2" range-step="0.05"
            value-pos="2" inverted="false">
            <default>1.0</default>
            <variable>HS_1</variable>
            <item>"0.2 | 3 | 0.2 "</item>
            <item>"0.4 | 3 | 0.4 "</item>
            <item>"0.6 | 3 | 0.6 "</item>
            <item>"0.8 | 3 | 0.8 "</item>
            <item>"1.0 | 3 | 1.0 "</item>
            <item>"1.2 | 3 | 1.2 "</item>
            <item>"1.4 | 3 | 1.4 "</item>
            <item>"1.6 | 3 | 1.6 "</item>
            <item>"1.8 | 3 | 1.8 "</item>
            <item>"2.0 | 3 | 2.0 "</item>
            <output file>/tmp/h1_out</output>
          </hscale>
        </frame>
      </vbox>
    </frame>
  </vbox>
</frame Green>
```

```
<hscale width-request="800" height-request="80"
  range-min="0.1" range-max="2" range-step="0.05"
  value-pos="2" inverted="false">
  <default>1.0</default>
  <variable>HS_2</variable>
  <item>"0.2 | 3 | 0.2 "</item>
  <item>"0.4 | 3 | 0.4 "</item>
  <item>"0.6 | 3 | 0.6 "</item>
  <item>"0.8 | 3 | 0.8 "</item>
  <item>"1.0 | 3 | 1.0 "</item>
  <item>"1.2 | 3 | 1.2 "</item>
  <item>"1.4 | 3 | 1.4 "</item>
  <item>"1.6 | 3 | 1.6 "</item>
  <item>"1.8 | 3 | 1.8 "</item>
  <item>"2.0 | 3 | 2.0 "</item>
  <output file>/tmp/h2_out</output>
</hscale>
</frame>
<frame Blue>
  <hscale width-request="800" height-request="80"
    range-min="0.1" range-max="2" range-step="0.05"
    value-pos="2" inverted="false">
    <default>1.0</default>
    <variable>HS_3</variable>
    <item>"0.2 | 3 | 0.2 "</item>
    <item>"0.4 | 3 | 0.4 "</item>
    <item>"0.6 | 3 | 0.6 "</item>
    <item>"0.8 | 3 | 0.8 "</item>
    <item>"1.0 | 3 | 1.0 "</item>
    <item>"1.2 | 3 | 1.2 "</item>
    <item>"1.4 | 3 | 1.4 "</item>
    <item>"1.6 | 3 | 1.6 "</item>
    <item>"1.8 | 3 | 1.8 "</item>
    <item>"2.0 | 3 | 2.0 "</item>
    <output file>/tmp/h3_out</output>
  </hscale>
</frame>
</vbox>
</frame>
</vbox>
<vbox>
  <frame>
    <hbox homogeneous="true">
      <timer visible="false" interval="200" milliseconds="true">
      <action function="save">HS_1</action>
      <action function="save">HS_2</action>
      <action function="save">HS_3</action>
      <action function="command">xgamma -quiet -rgamma `cat
/tmp/h1_out`</action>
      <action function="command">xgamma -quiet -ggamma `cat
/tmp/h2_out`</action>
```

```
<action function="command">xgamma -quiet -bgamma `cat
/tmp/h3_out`</action>
</timer>
<button image-position="2">
  <label>"Done"</label>
  <input file stock="gtk-cancel"></input>
  <action>EXIT:cancel</action>
</button>
</hbox>
</frame>
</vbox>
</vbox>
</window>'
```

```
export MY_GUI="`echo "$MY_DIALOG" | sed 's/##.*/'`"
gtkdialog -p MY_GUI
#rm -f /tmp/h1_out
#rm -f /tmp/h2_out
#rm -f /tmp/h3_out
```

# I decided to leave these for reference.



The PCLinuxOS Magazine  
Created with Scribus

## The PCLinuxOS Magazine Special Editions!

 A collage of six magazine covers for The PCLinuxOS Magazine. The covers are:
 

- Windows Migration Guide** (September 2013): Features a penguin looking through a magnifying glass.
- Enlightenment Special Edition** (May 2011): Features a penguin on a dark background.
- The KDE 4 SC Special Edition** (Fall 2010): Features several KDE desktop icons.
- Gtk Lightweight Desktops: Xfce & LXDE Special Edition** (November 2010): Features a yellow robot holding a pencil.
- Command Line Interface Intro Special Edition** (October 2010): Features a terminal window with green text on a black background.
- Openbox Special Edition** (March 2012): Features a penguin with a box of toys.

**Get Your Free Copies Today!**



# PCLinuxOS Puzzled Partitions

			8	5			4	
			2		1		6	8
				9		3		2
	4					1		6
		3				2		
2		1					9	
6		5		4				
1	7		6		2			
	8			3	7			

**SUDOKU RULES:** There is only one valid solution to each Sudoku puzzle. The only way the puzzle can be considered solved correctly is when all 81 boxes contain numbers and the other Sudoku rules have been followed.

When you start a game of Sudoku, some blocks will be prefilled for you. You cannot change these numbers in the course of the game.

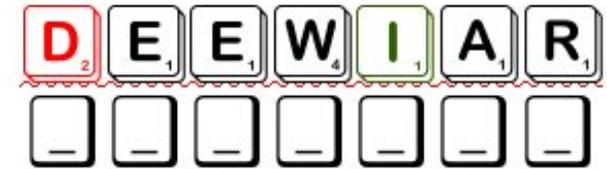
Each column must contain all of the numbers 1 through 9 and no two numbers in the same column of a Sudoku puzzle can be the same. Each row must contain all of the numbers 1 through 9 and no two numbers in the same row of a Sudoku puzzle can be the same.

Each block must contain all of the numbers 1 through 9 and no two numbers in the same block of a Sudoku puzzle can be the same.

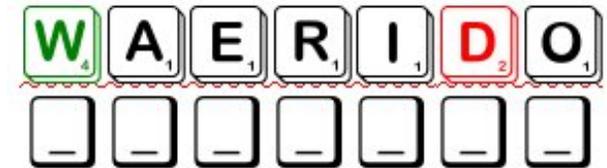


## SCRAPPLER RULES:

- Follow the rules of Scrabble®. You can view them [here](#). You have seven (7) letter tiles with which to make as long of a word as you possibly can. Words are based on the English language. Non-English language words are NOT allowed.
- Red letters are scored double points. Green letters are scored triple points.
- Add up the score of all the letters that you used. Unused letters are not scored. For red or green letters, apply the multiplier when tallying up your score. Next, apply any additional scoring multipliers, such as double or triple word score.
- An additional 50 points is added for using all seven (7) of your tiles in a set to make your word. You will not necessarily be able to use all seven (7) of the letters in your set to form a "legal" word.
- In case you are having difficulty seeing the point value on the letter tiles, here is a list of how they are scored:
  - 0 points: 2 blank tiles
  - 1 point: E, A, I, O, N, R, T, L, S, U
  - 2 points: D, G
  - 3 points: B, C, M, P
  - 4 points: F, H, V, W, Y
  - 5 points: K
  - 8 points: J, X
  - 10 points: Q, Z
- Optionally, a time limit of 60 minutes should apply to the game, averaging to 12 minutes per letter tile set.
- Have fun! It's only a game!



Double Word



Triple Word



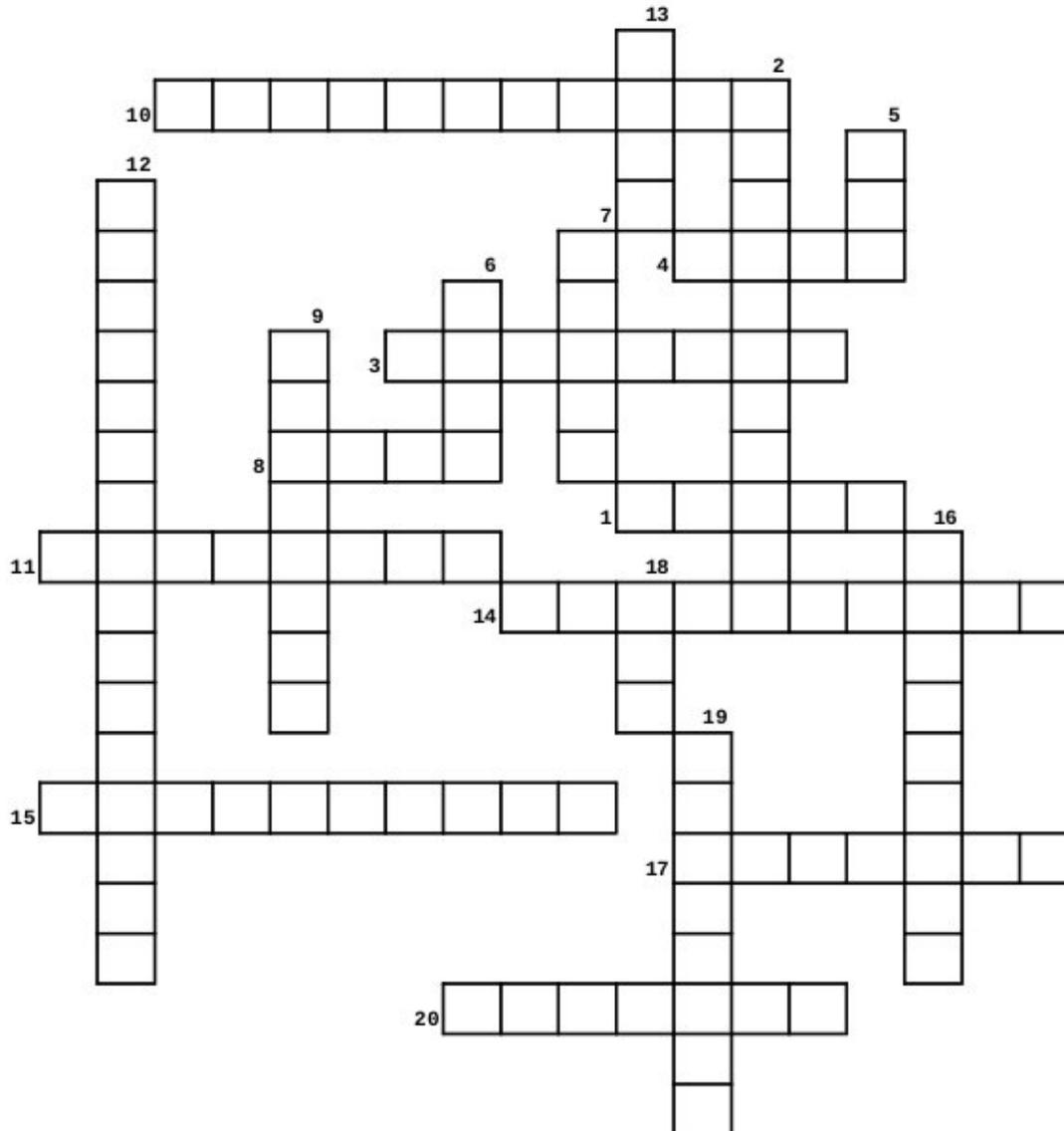
Possible score 290, average score 203.

Download Puzzle Solutions Here



# PCLinuxOS Crossword Puzzle: June 2015

## Camping



1. nice to stay in
2. will keep you warm while you snooze
3. a good place to hike up and down
4. if you don't stay in a cabin, this will be good
5. to shade your eyes or keep your head warm
6. a good item for mountain climbing
7. paddling down the river
8. fishing may be good here
9. animals you see
10. good to wear to protect your feet
11. a nice snack
12. for help with the mosquitoes
13. you may see this at night
14. park your trailer here
15. just in case you need some light
16. have this on hand to protect your skin
17. keep this full of water
18. you need this to locate the sights
19. a good container for your camping gear
20. to know which way you are going

[Download Puzzle Solutions Here](#)



# Camping Word Find

I F J W S M K J M H N W N A T U R E M K B N T X E U E E W H  
 L E K A L P O J S G S H X A U Y D P M A C W R X O I G I O O  
 O K H Q H W B P R F U D H F S H J T C P S A J E G C B L E H  
 H A E F B W O S B V K K J N K Q M I B F S A P X T S I G H C  
 K V Q L S I S J P T C V G R N Q D A I B T T W K H N S K R S  
 K N P X T C E F W O X A U E T Y A S P H O D O P C H A N X S  
 N T S L H T O Z M G E S E H T H H J N B O V N U Q A K L R W  
 A F O Y P Y O M J D Q T N C E I C J U A B O I K Y K B K R T  
 P N W C L U A B P B N A A L N N P D A W G X R C P L N M N C  
 S K G P M H R D R A W R W G A V A I Y H N P G P F I K F Y N  
 A M K C K L T C C E S S R L T T Y P D X I G G N M U J T O T  
 C L A F W P E F Z J T S E L P A C R J K K P A N C M D R C J  
 K Z S R O O Y G T W M A G T E H D E G L I G B I S A H A D D  
 T S E D D X Q R S E F A W N G F S U L R H X G O Z C P I K Q  
 T H C G F S B P E O N C L E I L N E Q V V T N W O K Q L O O  
 K L C R O T K G N N Z T A L H K F Y M N S H I Y D P D M G P  
 J V W C T O M C G W E G U L A O L W Q K J G P M J I S I D A  
 M B B E N O A X T S Q C F E C F W A U E K I E D Y H B X Y M  
 O W J P M B E E R E L D S P H Z R R W M P L E N M D W N H A  
 U W L Z I O E A S Y L T S E Y V A E T A N H L J R M T N V E  
 N T Z N S Q N C N T A B C R S A C W T L X S S H F C K N G S  
 T E P K S R O O D T U O V T N O O M R A J A K Z T O N D F J  
 A V Z U U T V N E Y L L O C F A O U U O W L K D E O S C K U  
 I I U U U D D P F S Z L X E G E G C X P I F Z J K P R B L P  
 N K S D B H A U T D B O P S C I W U R F C B T K X X O R A Q  
 S R K M E R K X K W J E O N A C Z Q T A Q P K V B O P R X J  
 P W N M K O R G U P N S Q I S G I F V S Q U P J N K K N P N  
 D O T V S Z Y M Q Z G D I M W J O G O Y S H Z S A E P C W U  
 A M W I L D L I F E F H Z T S U N S C R E E N L D L K A I V  
 W H E C U W D F I K A X H P D N U O R G P M A C O Y Z X Q H

backpack  
 boots  
 cabin  
 camp  
 campground  
 canoe  
 canteen  
 compass  
 fishing  
 flashlight  
 hammock  
 hat  
 hiking boots  
 insect repellent  
 knapsack  
 lake  
 lantern  
 map  
 moon  
 mountain  
 nature  
 outdoors  
 park  
 rope  
 scenery  
 sleeping bag  
 stars  
 state park  
 sunscreen  
 tent  
 trail mix  
 walking  
 water bottle  
 waterfall  
 wildlife

[Download Puzzle Solutions Here](#)



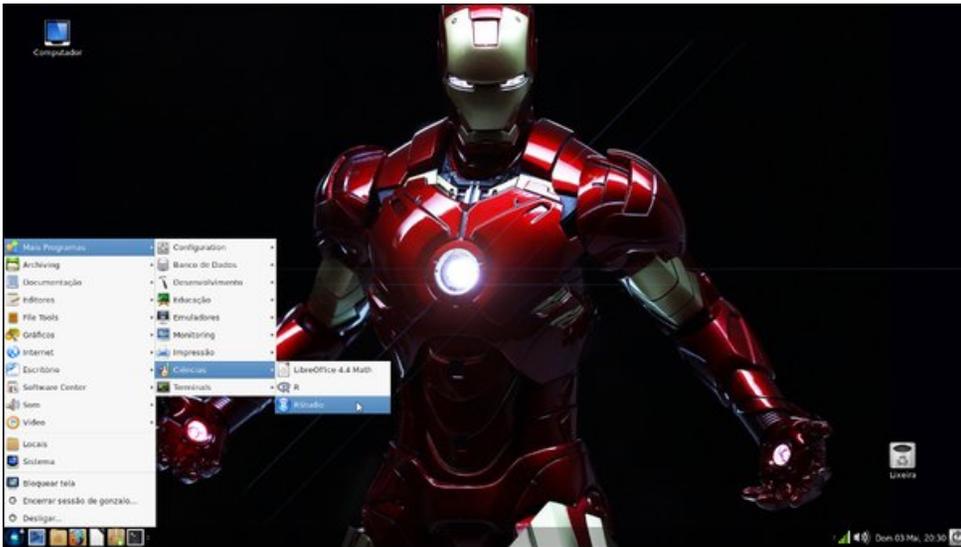
# More Screenshot Showcase



Posted by francesco\_bat, May 12, 2015, running KDE.



Posted by Hertz, May 1, 2015, running KDE.



Posted by Gonzalo\_VC, May 4, 2015, running Mate.



Posted by xredded, May 15, 2015, running KDE.