» **Gimp** Open source image-editing software you can get your teeth into

Gimp: Printing &

Gimp, CUPS and Gutenprint are a fellowship that can lead you on your quest for high quality prints – **Michael J Hammel** continues his graphics tutorial series.

Our expert

Michael J Hammel is a contributor to the *Gimp* project and the author of three books on the subject, including his latest, *The Artist's Guide to Gimp Effects.*

Ever since the birth of Gnome and KDE, printing support has been the bane of the Linux office desktop user. Advances in 2D and 3D graphics have been phenomenal thanks to the requirements of the special effects industry in Hollywood, yet with high-quality futuristic displays at our doorstep, we still settle for Stone Age prints. Until recently, the lack of a unified printing system has meant that desktop applications have either implemented their own printing features or relied on the lowest common denominators, often leading to low-quality prints or, even worse, no facility for printing at all. Even when we started to acquire good quality printer drivers and the necessary tools to use them on the desktop, colour management in Linux applications has been essentially non-existent.

Fortunately, that's started to change in a big way. The Linux printing scene has seen a wealth of printer driver support from both open source and commercial providers. As a result, the under-the-hood printing system has evolved into a system of libraries and utilities that desktop systems such as Gnome and KDE can harness – all tied together through the magic of CUPS.

Drivers, spoolers, filters and printers

Before we dive into how to use all of this with *Gimp*, it's important to have a basic understanding of how printing works under Linux. Printers come in two forms: PostScript and non-PostScript. PostScript is a device-independent language that describes the data to be printed. PostScript-capable printers can understand this language directly, whereas non-compatible printers must have data converted into a custom printer language. Getting files into PostScript in the first place is a multi-step process that involves a series of software packages.

The Linux printing environment is made up of four levels of software: drivers, spoolers, filters and applications. Printer drivers form the backbone of printing; they send data to the printer using a language the printer will understand. Some Epson printers, for example, use a language called ESC/P while other manufacturers use a language called PCL. The data must already be in that language when the driver receives it, however; the drivers are not responsible for doing any data conversions.

CUPS and Gutenprint

Printer drivers for Linux come primarily from the *Gutenprint* and/or *Ghostscript* projects, although HP have their own set of drivers (known as the HPIJS) that they actively support. Other vendors, including Epson, offer drivers for their printers as well. All of the drivers are intended to be fed data to print via the various Linux print spoolers.

Drivers are at the bottom of this four-layer software stack for printing. At the opposite end are applications, with spoolers just below. When an application is directed to print a file, it sends it to a print spooler, which queues it up to be processed later (when the spooler is not busy with other print jobs). There are many spoolers available for Linux, including LPR (Line Printer Remote), LPRng and CUPS (Common Unix Printing System). CUPS is becoming the standard for spoolers on Linux distributions for a variety of reasons, although none of these make much difference to end users. It's important to note that you'll be working with CUPS to set up your printers.

Once a spooler has a file queued for printing, it runs the file through one or more filters that convert it into various formats. The filtering process is better know as a 'RIP' (Raster Image Processing). In some cases the files need to be converted into PostScript first, then into a printer-specific format before being sent to the printer driver. In other cases the file may already be in PostScript format, or may not need to be converted to another form before being sent to the printer driver.

Filters can come from a number of software packages. The most widely used set of filters come from the *Foomatic* package, but you won't need to worry about them much because you'll get the ones you need when you install CUPS.

With all that software ready and under the hood, it's time to configure your printer. As well as checking the OpenPrinting website (**www.linuxfoundation.org/openprinting**) to see if the printer is supported, you'll also want to install CUPS. CUPS is both a spooler and a configuration tool. It allows you to configure your printers whether they're directly connected to your system, located on your local network or connected to another system elsewhere

» **Last month** We saw how selection mixes can pull complex subjects from images.

colour

on your network. There's a web-based config tool available, along with desktop-specific tools for both Gnome and KDE.

CUPS uses one or more filters to rip data into a format that can be used by a printer driver. This spooler package is designed to hide the filtering process, so users don't have to know whether a particular file is being converted before printing. All you need to know is what printer model you have and how it'll be connected to your system.

CUPS also uses PPD (PostScript Printer Description) files, which describe the features of printers to applications that need to use them. Despite the name, these files are used for describing both PostScript and non-PostScript printers. Again, in most cases you won't need to acquire these files if CUPS and *Gutenprint* support the printer.

Gutenprint includes drivers for a huge number of printers. However, because the package is under constant development, the version of *Gutenprint* on your Linux distribution may not be up to date. Fortunately, the version available from the OpenPrinting website is always matched against its database of supported printers, so if the database says the printer is supported but your version of *Gutenprint* doesn't have it, simply replace it with the new version from the OpenPrinting site.

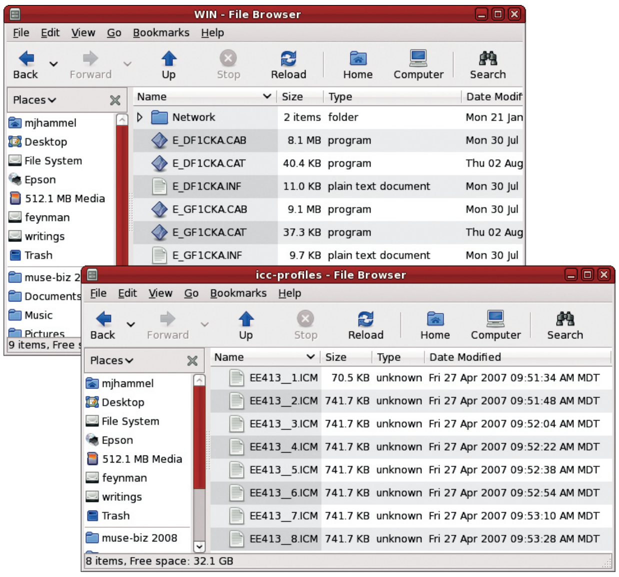
Colour management

Colour management is the process of matching colours between input devices (such as scanners and cameras), editing tools (such as *Gimp*) and output devices (such as your monitor or printer). Each input or output device has a range of colours it can work with, but you can usually edit a wider range with your applications. The goal of colour management software is to map colours from the input and editing phases to the range of colours that can be displayed on the output device in order to help assure the result is visually identical to the source.

Under Linux, colour management in applications is primarily handled through a package known as *LittleCMS*. This is another of those packages you don't need to worry about, because your application will install it if it needs it. Applications use *LittleCMS* in conjunction with ICC colour profiles, which are files describing the colour capabilities of a particular device, sometimes under specific conditions. For example, most cameras have their ICC profiles embedded in the images they take. This is why you sometimes see a message pop up in *Gimp*, telling you the image is in a certain colour space and asking if you want to convert it to a different one. Printers, on the other hand, often deliver their profiles via setup disks and usually provide one profile for each of the different types of paper they can print on to.

For printers under Linux, you may run into problems trying to find the ICC profile files on the hardware's setup disc. In my case, I was able to find the profiles for my new Epson Stylus Photo R280 buried in CAB archive files on the disc. You need to install the *Cabextract* utility to unpack these files. Do this by copying the CAB file to a directory and running:

```
cabextract filename.CAB
```



Name	Size	Type	Date Modif
Network	2 items	folder	Mon 21 Jan
E_DF1CKA.CAB	8.1 MB	program	Mon 30 Jul
E_DF1CKA.CAT	40.4 KB	program	Thu 02 Aug
E_DF1CKA.INF	11.0 KB	plain text document	Mon 30 Jul
E_GF1CKA.CAB	9.1 MB	program	Mon 30 Jul
E_GF1CKA.CAT	37.3 KB	program	Thu 02 Aug
E_GF1CKA.INF	9.7 KB	plain text document	Mon 30 Jul

Name	Size	Type	Date Modified
EE413__1.ICM	70.5 KB	unknown	Fri 27 Apr 2007 09:51:34 AM MDT
EE413__2.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:51:48 AM MDT
EE413__3.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:52:04 AM MDT
EE413__4.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:52:22 AM MDT
EE413__5.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:52:38 AM MDT
EE413__6.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:52:54 AM MDT
EE413__7.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:53:10 AM MDT
EE413__8.ICM	741.7 KB	unknown	Fri 27 Apr 2007 09:53:28 AM MDT

8 items, Free space: 32.1 GB

› For my Epson R280, there were two CAB files that contained identical binary ICM files. I unpacked one CAB file and saved only the ICM files.

I ran the **file** command on all files extracted from the archive and found that the ICM files were ICC colour profiles. I copied these to a directory where I can keep them safe (with my backups). I found multiple profiles and had to open them in *Gimp* to identify their purpose, which turned out to be one profile for each type of paper the printer supported.

Install LPROF

Once you have your printer configured, have the printer ICC profile files to hand and prepare to use colour management under Linux. We're not quite ready to do that yet because we don't have an ICC monitor profile, so first we need to create one. We can do that easily using *LPROF*, but that package isn't available for all distributions and requires a lot of effort to compile for those it doesn't support. This section is for those who have problems compiling *LPROF* on their systems. For those users with a distribution that already has *LPROF*, such as Ubuntu, skip to the next section to see how to use it.

First, download a copy of the latest version of *LPROF* from **http://lprof.sourceforge.net**. I'm working with version 1.11.4. This version has multiple requirements, the most important of which are:

» *Qt 3*

»

Specialised hardware

If the nature of your work means you need that extra degree of precision in your profiles, you should check out *ArgyllCMS* (**www.argyllcms.com/doc/ArgyllDoc.html**), a colour management system for Linux similar to *LittleCMS*. The documentation for *ArgyllCMS* lists a variety of specialised hardware that can be used for calibrating monitors and other devices.

»
- » *Vigra*
- » *LittleCMS* v1.16

The latest versions of most Linux distributions are likely to default to *Qt 4*, but they probably also have *Qt 3* available to install manually. On Fedora 9, I ran the following to get it:

```
sudo yum -y install qt3 qt3-devel
```

Vigra, (Vision with Genetic Algorithms) is a computer vision library. I installed both the runtime and development libraries for this package:

```
sudo yum -y install vigra vigra-devel
```

The most troublesome package of the three will be *LittleCMS* (**www.littlecms.com**), the latest version of which is 1.17. You can't install the prepackaged 1.16 into system directories in parallel with newest version, so you need to download the source code for v1.16, compile it and install in a separate directory. When I run into issues like this, I install the specialised package under **/usr/local/<packagename>**. I run whichever tools require this package with a shell script to make sure that the package is found before others with similar names or updated versions are found in the normal system directories.

Again, be sure you've downloaded the source for version 1.16. To compile *LittleCMS* and install it under its own directory under **/usr/local**, I run the following commands:

```
./configure --prefix=/usr/local/lcms
make
sudo make install
```

With the prerequisites installed I can now compile *LPROF* and install it under its own directory under **/usr/local**:

```
export PATH=/usr/local/lcms/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/lcms/lib:$LD_
LIBRARY_PATH
python scons.py cxxflags="-I/usr/local/lcms/include"
PREFIX=/usr/local/lprof
sudo python scons.py install
```

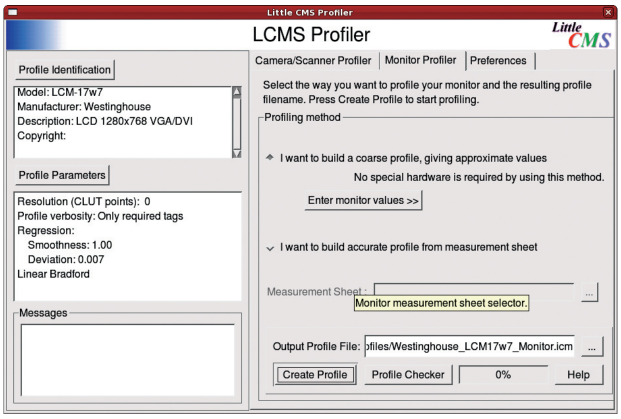
Management with LPROF

To run *LPROF*, I use this simple script:

```
#!/bin/bash
export LD_LIBRARY_PATH=/usr/local/lcms/lib:$LD_
LIBRARY_PATH
export PATH=/usr/local/lprof/bin:$PATH
    lprof
```

LPROF enables you to create ICC profiles for scanners, cameras and monitors. You need to use specialised hardware to get highly accurate profiles, but we can get a rough estimate for a monitor just using *LPROF*.

You can make a coarse approximation of a monitor profile just using your eyes. To begin, you should be in a dark room with the monitor's contrast control set to its highest level. If you can't be in a completely dark room, try placing a dark-coloured cloth over the monitor and your head while you're doing this. You might look a little silly, but you'll get better results. Don't leave the cloth over the monitor (especially a CRT monitor) for very long because it may overheat.



Model: LCM-17w7
Manufacturer: Westinghouse
Description: LCD 1280x768 VGA/DVI
Copyright:

Resolution (CLUT points): 0
Profile verbosity: Only required tags
Regression:
Smoothness: 1.00
Deviation: 0.007
Linear Bradford

Select the way you want to profile your monitor and the resulting profile filename. Press Create Profile to start profiling.

I want to build a coarse profile, giving approximate values
No special hardware is required by using this method.

I want to build accurate profile from measurement sheet

Output Profile File: ofiles/Westinghouse_LCM17w7_Monitor.icm

› The *LPROF* window (which titles itself as *LittleCMS* Profiler) has multiple tabs on the right. Start with the Monitor Profile tab.

The printing ecosystem

Find out if your printer is supported, get the latest news and learn more about printing under Linux by visiting the OpenPrinting section at the Linux Foundation's website: **www.linuxfoundation.org/en/OpenPrinting**.

Coarse approximations

The process of creating a coarse approximation profile is explained on the Monitor page of the *LPROF* documentation (**http://lprof.sourceforge.net/help/monitor.html**), but this page is a little hard to follow, so I'll walk you through the process. The *LPROF* window opens with a set of tabs on the right. In the Monitor Profile tab you'll see that the first option under Profile Method is to create a coarse profile. Click the Enter Monitor Values button to open the Monitor Values dialog. Click the Set Gamma And Black Point button under the Gamma entry. The window that opens has a small window labelled 'Gamma = 2.2' and a larger window to the right that's labelled 'Gamma And Black Level'. You'll see a slider at the the bottom, along with some other buttons.

Adjust the slider so that the Gamma = 2.2 window shows the horizontal lines and the box around them as close to a solid grey as you can make them. The lines probably won't go away when you do this; the trick is to stare at the box so your vision goes slightly out of focus and make sure that the grey gradient you see is as smooth as possible. This adjustment sets the gamma level for your monitor. Your monitor profile tells software to adjust images so they display as if your monitor were at 2.2.

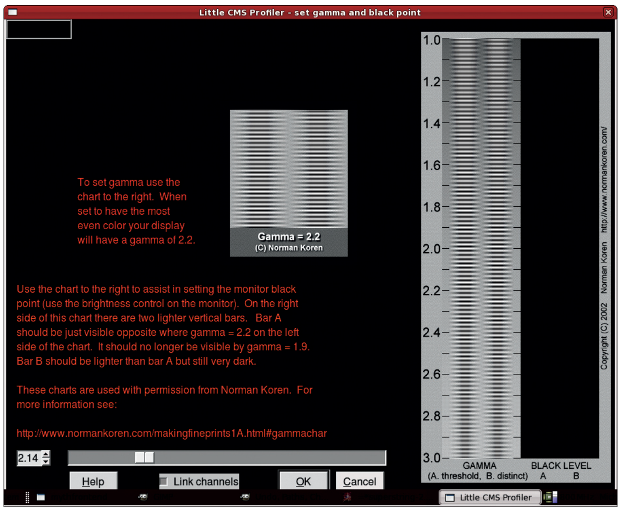
After setting the gamma level, you need to adjust the Brightness control of your monitor, which may also be labelled 'Black Point' (as it is on my Westinghouse LCD monitor). Adjust it so that the bars in the Black Level window on the right are barely visible across from 2.2 on the scale just to the left and disappear completely from view by 1.9. After making the Brightness or Black Point adjustment on your monitor, you may need to adjust the Gamma slider again. Repeat this process a few times until you're satisfied with the result.

You may need to click the Link Channels button to adjust the Red, Green and Blue channels separately. With each channel, try to adjust the gamma first, then adjust the Brightness on the monitor. Close the window by clicking on the OK button and then close the Rough Monitor Values window.

Create a new profile

Back in the *LPROF* main window, you'll see the Profile Identification button in the top-left. Click on this to fill in the monitor make and model information, which is often found on a label on the back of the monitor.

At the bottom-right of the *LPROF* window, just above the Create Profile button is the Output Profile field. Type in the path and name of the ICC output file, then click on Create Profile. You can check the profile using the Profile Checker button. Information on how to use this is found on the Profile Checker page of the *LPROF* documentation.


Little CMS Profiler - set gamma and black point
Gamma = 2.2
(C) Norman Koren
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
GAMMA
(A. threshold, B. distinct)
BLACK LEVEL
A
B
Copyright (C) 2002 Norman Koren http://www.normankoren.com/
2.14
Help
Link channels
OK
Cancel
Little CMS Profiler

To set gamma use the chart to the right. When set to have the most even color your display will have a gamma of 2.2.

Use the chart to the right to assist in setting the monitor black point (use the brightness control on the monitor). On the right side of this chart there are two lighter vertical bars. Bar A should be just visible opposite where gamma = 2.2 on the left side of the chart. It should no longer be visible by gamma = 1.9. Bar B should be lighter than bar A but still very dark.

These charts are used with permission from Norman Koren. For more information see:

http://www.normankoren.com/makingfineprints1A.html#gammachar

› Use the Gamma/Black Point window to visually configure your monitor profile without specialised hardware.

Using profiles is very straightforward, but interpreting the results and adjusting an image accordingly takes some experimentation. *Gimp 2.4* introduced support for ICC profiles in the Colour Management page of the Preferences dialog. This is where you select the profiles for your printer and the monitor profile that we just created with *LPROF*.

The Mode of Operation controls how you will view your images in the Canvas window. Setting this to Colour Managed Display will cause *Gimp* to display images using your monitor profile (which we'll configure in a moment), while setting it to Print Simulation will cause *Gimp* to display the image using the printer profile. This is what you want to use when you want to preview your image to check for out-of-gamut colours. These are colours that can't be reproduced by your printer given the printer profile you'll be using. The purpose of the RGB and CMYK settings isn't clear from *Gimp* documentation, but for now you can leave these set to None.

Interpreting profiles

Use the Monitor Profile setting to select the profile you created using *LPROF*. The Print Simulation profile is used to select the profile for your printer. Notice that I have multiple profiles and the *Gimp* dialog for selecting the profile provides information about the profile. For my Epson printer, I have profiles set up for different papers, most of which were created using a White Point setting of 5000K (which is shown here as D50).

Gimp has multiple methods available for interpreting profiles. These can be set with the Display Rendering Intent (for interpreting how images are shown on the Canvas) and Softproof Rendering Intent (for how printed images will be previewed on the Canvas) options in the Preferences dialog. The various options available are quite technical and the Perceptual option is probably the best for most home and small business use. Professional designers can read the *Gimp* help documentation for details on the other options.

The File Open Behaviour option determines what *Gimp* will do with images it opens that have ICC profile data embedded in them. Cameras often embed profiles in the images they create. There's little reason not to set this to Convert To RGB Workspace; leaving it as 'Ask What To Do' will cause a prompt to be displayed asking you what to do when you open images with embedded profiles.

The most important option in the Preferences is the Mark Out Of Gamut setting. With this option enabled, and the Mode of Operation set to Print Simulation, the Gimp Canvas window will highlight any pixels that won't print properly with a specified colour (which defaults to grey). This enables you to see which pixels will have to be converted to another colour by the printer. It will, in effect, tell you how different your image is likely to be from what you see when you're working on the image on screen.

With out-of-gamut colours marked, you may wonder how you can fix an image to print properly (without having the printer convert them to different colours). Unfortunately, there is no fix-all option. For my images, I adjusted the different Colour Channels in the Curves dialog by moving the end points of the graph away from the corners. When you do this with Preview enabled in the Curves dialog you can watch the grey, out-of-gamut colours change. In this case, the fix took only a few seconds, but you may find that the Curves dialog is not sufficient for your needs. In that case, you'll need to experiment with other tools in the Colour menu.

Printing

With your image colour corrected you're ready to print. Remember we used CUPS and *Gutenprint* to set up our printer. *Gimp* has its own print dialog built into it. *Gutenprint* also provides a separate print plugin for use with *Gimp*. With both enabled, you can choose which is more appropriate for your needs.

Gimp's built-in Print window offers a print preview, but it doesn't make clear whether the preview is colour-adjusted based on ICC profiles. Still, for most home printing, you'll probably be happy with the preview it provides.

Gutenprint's Print window offers much finer control over the print process itself. The print preview is purposely very low quality to make it easier to drag it around the page for alignment. This preview does not make use of your printer's ICC profiles. What's most important about the *Gutenprint* dialog is that you can configure any printer you might have connected to your computer for use with *Gimp*, even if that computer isn't configured through CUPS. This is useful, for example, if you have a printer used specifically for *Gimp* prints, but not shared with others in your household or business.

Summary

Printing support under Linux has become easily good enough for the home and small business user due to advances in *Gutenprint*, CUPS and the availability of drivers from commercial printer vendors. Colour management under *Gimp* is still a work in progress, but the 2.4 release has brought us closer to colour Nirvana with the ability to work directly with ICC profiles. While there's still work to be done with regard to correcting of out-of-gamut issues, we can at least know when an image won't fit and can manually work at colour correction to fit the image into gamut for the printer. LXF

To convert or not to convert

Unless you're a graphic designer working with a client who has specific colour requirements, you probably won't worry much about colour management and ICC profiles when using *Gimp*. If you don't care about such things, it won't matter if you convert images from their original colour space. If you do care, you should convert the image before working on it. Be certain to save your work into a different file so you can maintain the colour profile information in the original file.

» Next month We'll use quick colour fixes to adjust tints and whiten a smile.