

## Digital Cameras on Linux

# Digital Picture Box

Digital cameras continue to replace traditional cameras in today's homes. The typical Linux approach to transferring images from the camera to the hard disk is gPhoto or the kernel-based USB mass memory module. Graphical front-ends like digikam promise enhanced usability.

BY ANDREA MÜLLER



**B**uying films, having them developed, sorting the prints, buying photo corners and mounting the prints in albums. All of these time-consuming tasks are now things of the past for amateur photographers who have moved over to a digital camera from traditional wet film equipment. Those who have made the move, face different issues, such as finding the best software to run on their computer to transfer the digital images from the camera to their home PC. The answer for Linux depends on the make and model of the digital camera.

Devices that attach themselves to the system as though they were hard disks provide the easiest approach, allowing users to enter a simple *mount* command to give access the device. If you have a camera that uses a special protocol to talk to the PC, and your camera is a recent model, you still stand a good chance of being to access the camera using a software produced by the gPhoto project [1]. This article describes both approaches, and introduces software

specifically designed for interfacing with digital cameras.

## Mounting a Camera

Some cameras, such as the Aito or Casio Exilim series, attach to a system as a SCSI hard disk, just like a USB memory stick is connected. You can use the *mount* command to mount the camera device, checking your */var/log/messages* file to discover the right device:

```
Feb 1 19:27:18 aries kernel: SCSI device sdb: 20208
512-byte hdwr sectors (10 MB)
Feb 1 19:27:18 aries kernel: sda: Write Protect is off
Feb 1 19:27:18 aries kernel: sda: sda1
```

The following command, issued by root, mounts the camera it detected as *sda1*:

```
mount -t vfat /dev/sda1
/mnt/usb -o umask=0
```

in */mnt/usb*. The devices are formatted

with the VFAT filesystem, typically used by Windows 9x. As this filesystem does not have any permissions, the administrator can assign permissions using the *umask* option. *0* is the most permissive value and grants read and write access to any camera directory for any user.

An entry in */etc/fstab* will allow non-privileged users to mount the camera. Mount options allow a more restrictive approach to permissions, such as assigning a *usbusers* group permissions to delete files on the camera. After creating the group, and defining the mountpoint, the entry might appear as follows:

```
/dev/sda1 /mnt/usb auto users,
noauto,gid=333,umask=002,
showexec 0 0
```

The *gid* entry is followed by a numerical ID, as stored in the */etc/group* file. All members of the group have read and write access to the camera. The *users* entry, in contrast to *user*, allows non-privileged users to unmount the camera,

despite not having mounted the camera themselves.

*showexec* works around a problem that stems from the fact that VFAT does not have a rights system. Users need to be assigned execute permissions in the *umask*, to allow them access to the directory structure. One side effect is that normal files are tagged as being executable. Some Linux programs, such as older versions of the Rox file manager, refuse to display a preview for executable images. You can use *showexec* to work around this, as it emulates Windows behavior. Files with the suffixes *.exe*, *.bat*, and *.com* are tagged as executable, but no others (see Figure 1). The *fstab* entry allows normal users to mount the camera in */mnt/usb* using *mount /dev/sda1*.

Suse Linux users can avoid this manual work. The *hotplug* daemon automatically creates an appropriate *fstab* entry, as soon as a USB mass memory device is attached.

You can use more or less any file manager to transfer the image files from the camera device to your hard disk. If you prefer a specialist program, try *Digikam* [2]. This is the only camera front-end that works with *gPhoto 2* and also supports USB mass memory devices.

## The gPhoto Project

*gPhoto 2*, which reached version 2.1.4 in January 2004, consists of two components: the *libgphoto2* library, which supports more than 400 camera types, and the *gphoto2* command-line program, which allows users to access supported cameras.

The project has its roots in *gphoto0.4.x*. The older version had an integrated graphical front-end and a monolithic architecture. *gphoto2* has a modular structure, and was completely re-written by its developers. The first stable version was released for downloading in February 2002. New versions of the software are released every few months, with beta versions and pre-releases to stop the gap between stable releases, and allow users to test new cameras.

As the versions included by distributions tend to lag behind the current state of development, users with new models will need to compile the sources.

However, this will also require you to re-compile the graphical front-end, and *gtkam*, as this tool will not typically run with the later version if it is linked against an older library.

Before you go to all this trouble, you should enter the *gphoto2 --list-cameras* command to find out which devices are supported by the version supplied with your distribution. If you draw a blank for your model, you can surf to [3] to check the compatibility list for the current *gphoto2* version.

More recently, *gPhoto 2* has improved support for new models. Many of today's cameras use the Picture Transfer Protocol (PTP) [4], which ensures a simple file transfer between PTP devices. This removes the need for special drivers, and allows any computer to exchange data with any other PTP-aware devices without installing additional software.

These devices do not need to be limited to cameras and computers; exchanging data with cellphones or printers is just as easy. PTP-cameras run "out of the box" on Windows XP and MacOS X.

## Enhanced Support thanks to the PTP Standard

Although the standard was published back in 2000, very few manufacturers were prepared to adopt it originally – Kodak and Sony were among the pioneers. Fortunately, other manufacturers have begun to appreciate the advantages, such as cost savings, as they no longer need to produce their own drivers.

*gPhoto 2* introduced PTP camera support in June 2002. It was fairly spartan at first, and just about capable of transfer-

ring image files to disk. Other commands, such as outputting lists of image files, led to reproducible I/O errors; GUI-based front ends only rarely worked. Stability improved dramatically with version 2.1.3. However, a camera explicitly mentioned in the list of supported devices will be better supported than a generic *USB PTP Class Camera*. This feature is still in the test phase.

## Permissions

If your camera is supported, you still have to deal with the problem of permissions, unless your distributor has taken care to allow users other than *root* to access the camera. USB cameras are detected by the system as entries below the mountpoint */proc/bus/usb*. This is where the USB device filesystem is located; the kernel generates the filesystem generically in a similar way to the */proc* pseudo-filesystem [5]. Permission changes are lost each time you reboot.

There are two ways to provide users access to the camera:

- Changing the mount options for the USB filesystem;
- Dynamically setting permissions using the *hotplug* daemon.

The first approach is the simpler of the two. The following */etc/fstab* entry

```
none /proc/bus/usb usbfs 2
devmode=0664,devgid=333 0 0
```

mounts in */proc/bus/usb* with read and write permissions for user and groups with the ID 333 (this is *usbusers* on our system). Users who are not members of this group have read access only.

These mount options are valid for any USB device and not just for the camera.

```

[andri@dboomtrain.localdomain: /home/andri]
[andri@dboomtrain andri]# mount -t vfat /dev/sda1 /mnt/usb/ -o umask=002,gid=333
[andri@dboomtrain andri]# ls -l /mnt/usb/dcm/100casio/
total 2240
-rwxrwxr-x 1 root usbusers 364848 Jan 31 2003 cimg0042.jpg*
-rwxrwxr-x 1 root usbusers 364038 Jan 31 2003 cimg0044.jpg*
-rwxrwxr-x 1 root usbusers 367606 Jan 31 2003 cimg0045.jpg*
-rwxrwxr-x 1 root usbusers 371584 Jan 31 2003 cimg0046.jpg*
-rwxrwxr-x 1 root usbusers 374121 Jan 31 2003 cimg0047.jpg*
-rwxrwxr-x 1 root usbusers 394741 Jan 31 2003 cimg0048.jpg*
[andri@dboomtrain andri]# umount /dev/sda1
[andri@dboomtrain andri]# mount -t vfat /dev/sda1 /mnt/usb/ -o umask=002,gid=333,showexec
[andri@dboomtrain andri]# ls -l /mnt/usb/dcm/100casio/
total 2240
-rw-rw-r-- 1 root usbusers 364848 Jan 31 2003 cimg0042.jpg
-rw-rw-r-- 1 root usbusers 364038 Jan 31 2003 cimg0044.jpg
-rw-rw-r-- 1 root usbusers 367606 Jan 31 2003 cimg0045.jpg
-rw-rw-r-- 1 root usbusers 371584 Jan 31 2003 cimg0046.jpg
-rw-rw-r-- 1 root usbusers 374121 Jan 31 2003 cimg0047.jpg
-rw-rw-r-- 1 root usbusers 394741 Jan 31 2003 cimg0048.jpg
[andri@dboomtrain andri]#

```

Figure 1: The mount option "showexec" copies Windows file permissions under Linux.

The hotplug daemon provides a more elegant approach to solving this problem. It monitors the system's USB ports, and recognizes any attached USB devices by their IDs, which are stored in files below `/etc/hotplug`. Depending on your distribution, the daemon parses either the `/etc/hotplug/usb.usermap` or `/etc/hotplug/usb/usbcam.usermap` file for the camera.

## Scripting

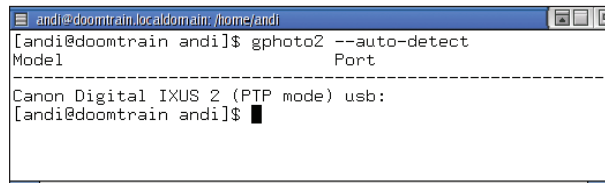
When the daemon connects to a device, it calls the `/etc/hotplug/usb/usbcam` script, which sets permissions for the attached camera. `libgphoto2` provides four sample scripts. If you built the program yourself, you will find them in the source code directory below `packaging/linux-hotplug`. Most distributions store them below `/usr/share/doc/libgphoto2-versionnumber`.

The four supplied sample scripts consist of: `usbcam.group`, where the administrator enters the group to be assigned the correct permissions, `usbcam.user`, which expects the name of the user to be granted permissions, `usbcam.x11-app`, which will automatically launch a gPhoto front-end, and `usbcam.console`.

The script requires PAM [6], which is installed by default on Red Hat or Mandrake Linux. It assigns all permissions for the camera to the user logged on to the console. If you build your own gPhoto version, you will need to update `usbcam.usermap` using the following command:

```
/usr/local/lib/libgphoto2/➤
print-usb-usermap > ➤
/etc/hotplug/usb/usbcam.usermap
```

If your distribution parses `usb.usermap`, you will need to delete any lines that



```
andi@doomtrain.localdomain: /home/andi
[andi@doomtrain andi]$ gphoto2 --auto-detect
Model                               Port
-----
Canon Digital IXUS 2 (PTP mode) usb:
[andi@doomtrain andi]$
```

Figure 2: gPhoto 2 recognizes newly attached cameras automatically by the USB connection.

start with `usbcam`, and then add the current camera ID to the end of the file:

```
/usr/local/lib/libgphoto2/➤
print-usb-usermap >> ➤
/etc/hotplug/usb.usermap
```

Setting up the camera like this allows normal users access. A graphical front-end is not necessarily required. The gPhoto 2 command-line tool provides a wide range of functions. The following syntax:

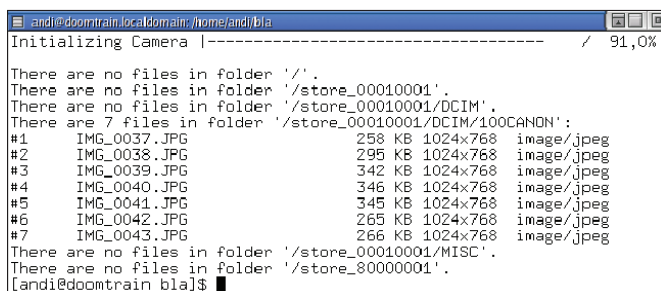
```
gphoto2 --auto-detect
```

shows our camera as detected by the program (see Figure 2).

## Command-Line Access to the Camera

Just a few flags are required for basic access. `-l` lists the directories, `-L` adds the filenames. Users can refer to the serial numbers in this output (see Figure 3 below) using the `-p number(n)` syntax, to filter specific files. `-P` refers to all the images found in the current working directory.

If you only want to transfer specific image files to the hard disk, you can specify these files as `gphoto2 -p 2,4,5` to download images 2, 4, and 5, or specify a range, as in `gphoto2 -p 2-4`. It is quite simple to exclude files from a range. For example, `gphoto2 -p 2-6,4` downloads the images 2, 3, 5, and 6, but not file number 4.



```
andi@doomtrain.localdomain: /home/andi/bla
Initializing Camera |-----| 91,0%
There are no files in folder '/'.
There are no files in folder '/store_00010001'.
There are no files in folder '/store_00010001/DCIM'.
There are 7 files in folder '/store_00010001/DCIM/100CANDN':
#1  IMG_0037.JPG      258 KB 1024x768 image/jpeg
#2  IMG_0038.JPG      295 KB 1024x768 image/jpeg
#3  IMG_0039.JPG      342 KB 1024x768 image/jpeg
#4  IMG_0040.JPG      346 KB 1024x768 image/jpeg
#5  IMG_0041.JPG      345 KB 1024x768 image/jpeg
#6  IMG_0042.JPG      265 KB 1024x768 image/jpeg
#7  IMG_0043.JPG      266 KB 1024x768 image/jpeg
There are no files in folder '/store_00010001/MISC'.
There are no files in folder '/store_80000001'.
[andi@doomtrain bla]$
```

Figure 3: gPhoto 2 displaying a list of the images on the camera.

## Intuitive Filenames

The program can do more than this. Digital cameras assign fairly non-intuitive filenames to images, but `gphoto2` can change the names when downloading files. To do so, specify the `--filename` parameter, followed by a name. `--filename` accepts typical

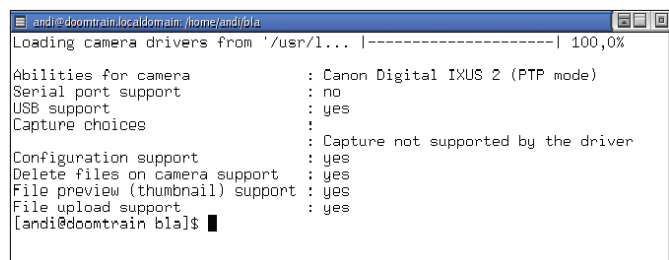
`date` format strings, such as `%y`, `%m`, and `%d` for year, month, and day. Also, `%n` specifies a number, `%C` specifies a file suffix, and finally `%f` specifies a filename without a suffix. `gphoto2 -P --filename birthday_john_%n.%C` downloads the photos from the camera and constructs a filename using the string `birthday_john_`, the image number, a dot, and the file suffix.

The parameters `-T` and `-t number(s)` download any or a selection of thumbnails for your photo collection. If you have a camera that can store audio clips in addition to images, you can specify `--get-audio-data number(s)` or `--get-all-audio-data` to access these files. To make room for new images, enter `--delete-files number(s)` and `--delete-all-files`.

The features that gPhoto 2 offers depend to some extent on the attached camera. `gphoto2 -a` will output the features of the attached device (see Figure 4).

Some cameras allow you to upload image files. The `-m` flag, followed by a name, creates a directory on the camera, and `gphoto2 -u filename` uploads files. The output from `gphoto2 -a` is not entirely reliable. Although the driver for our lab camera is not supposed to support `capture` according to the program, it is possible to remotely control the camera, and `gphoto2 --capture-image` will capture an image. Unfortunately, the `--capture-sound` option led to reproducible I/O errors in our Linux labs.

The `--shell` parameter allows users to navigate the camera directories, just



```
andi@doomtrain.localdomain: /home/andi/bla
Loading camera drivers from '/usr/l... |-----| 100,0%
Abilities for camera           : Canon Digital IXUS 2 (PTP mode)
Serial port support           : no
USB support                    : yes
Capture choices                :
                               : Capture not supported by the driver
Configuration support         : yes
Delete files on camera support : yes
File preview (thumbnail) support : yes
File upload support           : yes
[andi@doomtrain bla]$
```

Figure 4: The `-a` flag lists the camera's capabilities, telling you if the camera supports `thumbnails` for example.

like a local filesystem. Users can enter *gphoto2 --shell* to pop up the program's internal command-line. If you have ever used a text-based FTP client, you should have no trouble finding your way around.

### Additional Functions

*help* in the gPhoto2 shell lists the available commands, *help commandname* displays the syntax for individual commands. *ls* lists the files. *cd* changes to another camera directory. *delete file* deletes a file. *get file* starts downloading a photo to the current working directory, and *get-thumbnail* does the same thing with the thumbnail for the current image. If you open the gPhoto shell in a directory where you do not have write privileges, you can type *lcd directory* to change to another local directory. The *show-exif file* command is designed to display the meta-data for an

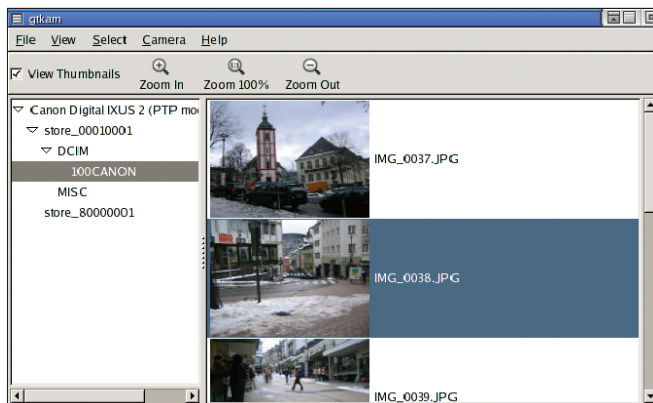


Figure 5: Clear-cut and functional: the GTKam GUI.

image file, but refused to work in our lab. *q*, *quit*, or *exit* quits the program's command line.

### Graphical Front-Ends

If you are unhappy with the command line, you can try one of the graphical front-ends for *gphoto*. There are several currently competing for the position of the amateur photographer's best friend.

The oldest contender is *gtkam* [7], which is a fairly staid affair (see Figure 5) that provides only basic functionality. Selecting *Camera | Add Camera | Detect* tells the program to access gPhoto 2's auto-recognition function. You can use a file manager style tree view to navigate the directory structure on a supported camera. GTKam displays images in the right-hand panel. The thumbnail view provides an initial over-

view. The *File* menu allows you to download all or selected images, and delete any images that you have successfully transferred to your hard disk.

*digikam* provides far more user convenience. Instead of deserting the user after downloading the image files, the tool also manages images on the hard disk. Images are organized in albums, which are in fact simple directories. The program will display the albums rather

EASY TO BUY • EASY TO SET UP • EASY TO SEE



one elephant...

Be a host with the most for the least using our world class web hosting services. From just £30 a year.



two elephants...

Do the business brilliantly with our EasyVserver. Launching your own web business has never been so affordable. Choose from 4, 8 or 16 Gigabyte options. There's no set up fee and web host manager makes managing multiple domains and email accounts easy.



three elephants...

Make a name for yourself with an unforgettable e-mail address from the UK's favourite domain name registrar. Why be boring? Give hotmail the cold shoulder. Give yahoo the old heave-ho. And be unique, such as [nothingon@all.tv](mailto:nothingon@all.tv), [rich@thirty.me.uk](mailto:rich@thirty.me.uk), [hrh@buckhouse.info](mailto:hrh@buckhouse.info), [tuff@thetop.org.uk](mailto:tuff@thetop.org.uk).

From only £3.99 a year.



four elephants...

Access email anytime, anyplace, anywhere with Easypost. That's our brilliant web mail package that enables you to pick up and send email using your domain name, making you look more professional. It's also anti-spam and anti-virus to ensure you don't pick up anything nasty along the way.



five elephants...

Log on in seconds and refresh your business image the easy way.

Count on us for big web benefits

Log on now and find out why 1/2 million customers already put us first.

Register at [www.easyspace.com](http://www.easyspace.com)

**Easyspace**   
your perfect partner for the web

than providing an overall view of the complete filesystem tree. Thumbnails are displayed in the right-hand panel (see Figure 6). You can drag & drop an image to copy or move it to another album in the tree view.

## Digikam, the Shooting Star

Digikam demonstrates an open approach when setting up a camera. Besides models supported by gPhoto 2, it also provides access to USB mass storage devices. You can select *USB Mass Storage* and enter a mountpoint (see Figure 7) to set up this type of device. Selecting *Camera | Connect* mounts the device in the file system.

Besides functions for downloading and deleting images, Digikam also has an upload function. To view images at full scale, you can optionally open your images in an external viewer, rather than in the internal viewer. KDE programs such as *kuickshow* and *kview* are examples of external viewers. A slideshow function, which optionally supports filename labels, rounds off the program.

The new version 0.6.0 may already have been released by the time this article appears. Version 0.6.0 will again have added functionality. Digikam uses a plug-in architecture in the new version, allowing developers to extend the program's functionality. Release Candidates 1 and 2 already had a feature-rich plug-in collection that provide the following extensions among others:

- Slideshow with effects;
- MPEG encoder which can use the *mjpeg-Tools* and *ImageMagick*, to create videos for VCDs, SVCDs, or DVDs;
- HTML Gallery, an extension of the Konqueror Gallery plug-ins;
- Comment editor;
- Photoprint Wizard to arrange multiple images on a single page;
- CD archival (needs K3b);

- Mail function with options for changing the size and compression rate of the files. Works with KMail, Sylpheed, and Mozilla.

With regards to managing and archiving digital images, Digikam has developed features that cater to user's every need.

*flphoto* [8] takes a different approach.

The program is really an image management tool that offers camera access as an add-on. The tool lacks a separate option for deleting images from the camera. The *Album | Import | Camera* dialog does provide a simple workaround, however. When you launch this item, FLPhoto autodetects the camera, and displays preview thumbnails of the images in the transfer window (see Figure 8).

The image management features themselves are very advanced. The program works with albums, which are actually text files containing the names of the images in the album. If you select *Image | Remove* to delete an image from an album, this does not delete the image file physically from your disk. The preview bar at the bottom of the window is more practical than the typical tree view, as it leaves more room for images on the desktop (see Figure 9).

If you prefer to use the same program for all of your file operations, you might prefer to stick with the KDE file manager, Konqueror, when transferring images from you camera to disk.

Thanks to the *camera Kioslave*, the KDE file manager tool can use the *libgphoto2* library to access any supported digital cameras, just as though they were a locally attached filesystem. Typing *camera:/* in the address line as the URL launches the auto-detect process, and tells Konqueror to display the camera's root directory.

## Specialists for Exif Information

Images from some digital cameras include so-called Exif (Exchangeable Image File Format) [9] headers, which store information such as a time-stamp, the focal distance, the flash mode, and the camera type. Users who want to retain this meta-information while processing images cannot use traditional image manipulation tools to do so, as the meta-infor-

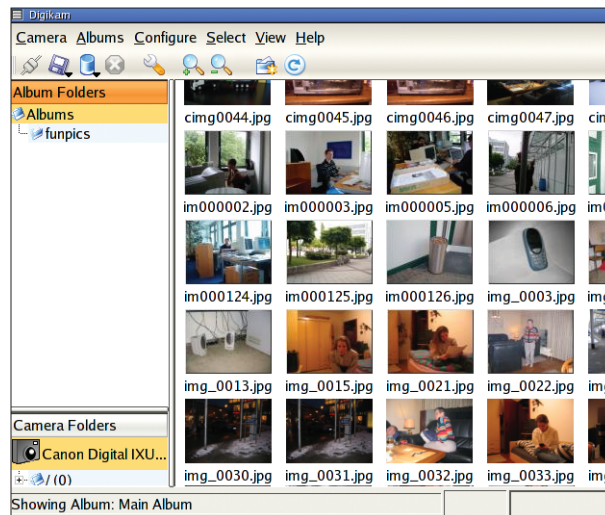


Figure 6: Tree view, camera view, and preview in Digikam.

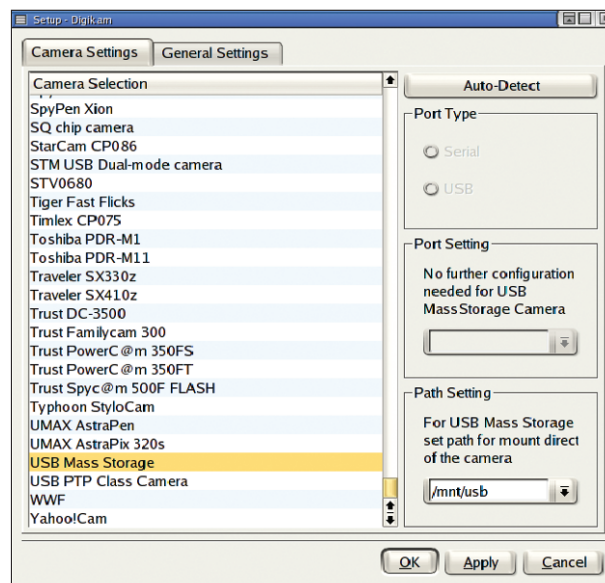


Figure 7: Digikam also supports cameras that the system detects as USB mass storage devices.

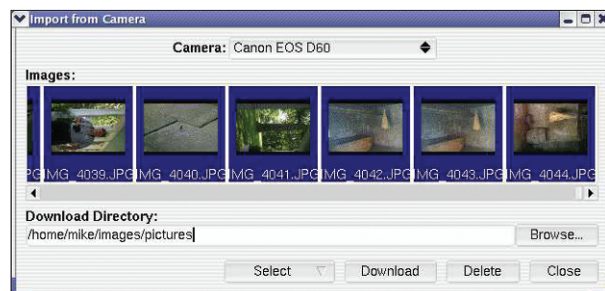


Figure 8: Importing images from a digital camera in FLPhoto.

mation is lost when you save the camera image on your hard disk.

The next version of Gimp looks set to provide an answer to this dilemma, as it can keep the Exif information. In the meantime, you should opt for a command-line tool called *jhead* [10]. *jhead* parses the Exif header, stores it, and then sends the image to another program where it can be edited. When editing is complete, *jhead* reinstates the Exif header. The following syntax would do just that:

```
jhead -cmd "jpegtran -rot 270
&i > &o" portrait_photo.jpg
```

*jpegtran* rotates the image 90 degrees to the left (-rot 90 will rotate through the same angle, but to the right). This is something that amateur photographers need quite often, if they take snapshots with camera in an upright position. *jhead* &i with the name of the original file. &o refers to a temporary file where the results will be stored. *jhead* inserts the Exif header into this file, and replaces the original image with the rotated image. As the program automatically assigns the right filename, you can use wildcards, like \*.jpg, to process multiple images in a single batch.

## Jpegtran Better than Imagemagick

Compared to *convert* from the Imagemagick package, the Jpegtran program used in our examples has the

advantage of creating loss-free jpeg files. If you use *convert*, storing a file will recompress the image and lose image data. Jpegtran is part of *libjpeg*, and should be available with most distributions.

The most straightforward solution for viewing Exif headers at present is a GTK program called *gexif*, which is available as a download from the *libexif* project homepage [11]. It uses tabs to display the individual fields, and even shows explanations for some types of content (see Figure 10). If you are not in a hurry, the next version of Digikam, 0.6, will have the same feature.

*exiftags* [12] is useful for displaying Exif information in the command line. It separates the output into camera and image specific parts. The same package also includes *exifcom*, which you can use to fill the UserComment field of the Exif header – this assumes that the camera actually sets this field. Calling the tool without any flags will display a comment, if available.

```
exifcom -w "my
comment"
imagefile.jpg
```

will set or overwrite the comment. If you want to prevent *exifcom* from prompting before actually overwriting, simply add the -f flag.

## Where There's a Will...

Some people still maintain that Linux and multimedia don't see eye to eye. They obviously have

INFO	
[1]	gphoto2: <a href="http://www.gphoto.org/">http://www.gphoto.org/</a>
[2]	Digikam: <a href="http://digikam.sourceforge.net/">http://digikam.sourceforge.net/</a>
[3]	List of gPhoto2 compatible cameras: <a href="http://www.gphoto.org/proj/libgphoto2/support.php">http://www.gphoto.org/proj/libgphoto2/support.php</a>
[4]	Free software with PTP support: <a href="http://ptp.sourceforge.net">http://ptp.sourceforge.net</a>
[5]	USB device filesystem: <a href="http://www.linux-usb.org/USB-guide/x173.html">http://www.linux-usb.org/USB-guide/x173.html</a>
[6]	PAM: <a href="http://www.kernel.org/pub/linux/libs/pam/">http://www.kernel.org/pub/linux/libs/pam/</a>
[7]	Gtkam: <a href="http://www.gphoto.org/proj/gtkam/">http://www.gphoto.org/proj/gtkam/</a>
[8]	flphoto: <a href="http://www.easysw.com/~mike/flphoto/">http://www.easysw.com/~mike/flphoto/</a>
[9]	Exif: <a href="http://www.exif.org/">http://www.exif.org/</a>
[10]	jhead: <a href="http://www.sentex.net/~mwandel/jhead/">http://www.sentex.net/~mwandel/jhead/</a>
[11]	gexif: <a href="http://sourceforge.net/projects/libexif/">http://sourceforge.net/projects/libexif/</a>
[12]	exiftags: <a href="http://johnst.org/sw/exiftags/">http://johnst.org/sw/exiftags/</a>

not looked at the digital camera world for quite a while, as there has been some tremendous progress in this area over the past few months.

The KDE project, Digikam, is a convincing example of this – a powerful, and comfortable package, that makes image processing on Linux a real pleasure. Even if you are out of luck, and Digikam does not support your camera, you can still resort to a USB-based card reader, which is guaranteed to be capable of transferring your snapshots to your hard disk. ■

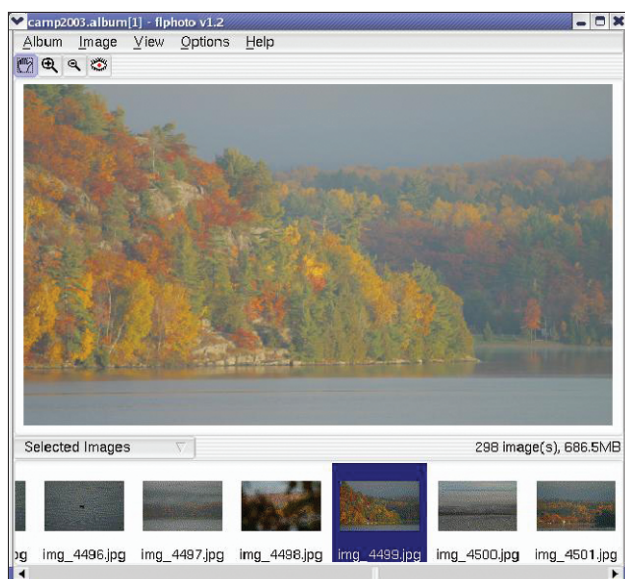


Figure 9: Preview window and image view in FLPhoto.

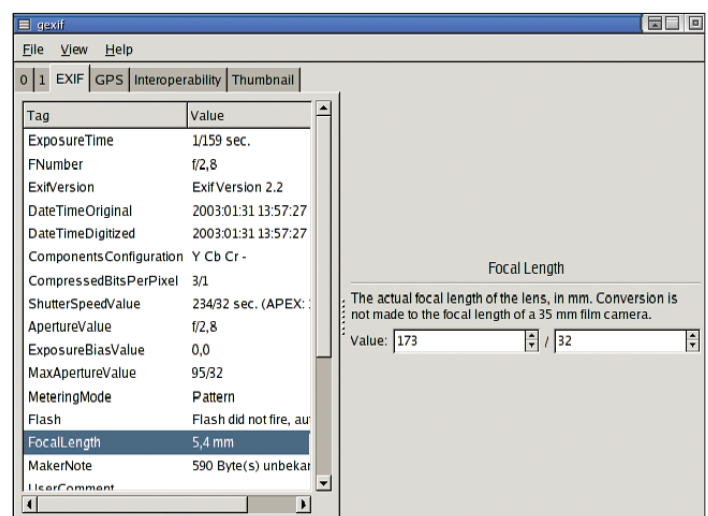


Figure 10: Gexif displays the full range of Exif information, from the camera settings to thumbnails.