

Effective netbook photo tools

PHOTO FRIEND

Find out how to turn your netbook into a fabulous photo tool.

BY DMITRI POPOV

Netbooks are usually advertised as machines suited for light-weight tasks like word processing and web browsing. But you can teach your humble netbook a few new tricks using the right software. For example, you can turn your netbook into a nifty digital photography companion.

Although installing GIMP and digiKam seems like a sensible thing to do, you should resist the temptation to load heavyweight applications onto your netbook. After all, you wouldn't want to use an underpowered netbook with a tiny screen and limited storage for heavy retouching and managing your entire photo library – especially when you are on the move.

THE AUTHOR
Dmitri Popov holds a degree in Russian language and computer linguistics. He has been writing exclusively about Linux and open source software for several years, and his articles have appeared in Danish, British, North American, German, and Russian magazines and websites.



Instead, you'd most likely use the netbook to pull photos from your digital camera, do some light tweaking, and upload them to your Flickr account.

gPhoto2

Let's start from the very beginning – getting photos from your digital camera onto the netbook. Virtually every netbook on the market comes with some sort of graphical application that allows you to download photos from your digital camera. However, if you want to fetch photos from the camera quickly and with minimum hassle, consider using the excellent gPhoto2 [1] command-line tool.

Besides photo transfer, gPhoto2 offers a few nifty features that allow you to manage the camera directly from the netbook (Figure 1).

gPhoto2 supports more than 1,000 camera models, so it works right out of the box with pretty much any camera on the market. Before you can start using gPhoto2, you have to let it detect and identify your camera by running the following command:

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

This should display the camera model, the transfer mode, and the port the camera is connected to – for example:

```
Canon Powershot SX100 IS ↗  
(PTP mode) usb:  
Canon Powershot SX100 IS (PTP mode) ↗  
usb:002,004
```

Being a command-line tool, gPhoto2 is rather straightforward to use. For example, you can download all photos from your camera using one simple command:

```
gphoto2 --get-all-files
```

This downloads photos to the current working directory, so make sure that you switch to the folder into which you want to download photos before you run the command. If your camera supports flagging already downloaded images, you can use the following command to fetch only the newest photos:

```
gphoto2 --get-all-files --new
```

These two commands are all you need to load photos off your camera, but

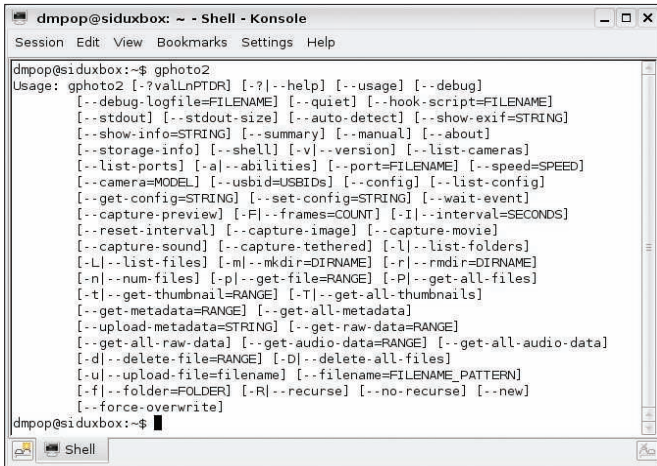


Figure 1: gPhoto2 allows you to transfer photos and control your digital camera.

gPhoto2 has a few other clever tricks up its sleeve. The most interesting one is the ability to remotely control your camera from the connected computer. This feature can be handy in many situations. For example, using the netbook as a remote trigger allows you to minimize camera shake, which can be particularly useful when shooting in low light conditions. To trigger a shot with gPhoto2, use the following command:

```
gphoto2 --capture-image
```

This fires a shot without making any adjustments, but you can also specify the aperture, shutter speed, ISO, and other settings with the `--set-config` command, provided this feature is supported by your camera. With the use of the `gphoto2 --list-config` command, you can see what parameters you can use with your camera. The `--interval` and `--frames` parameters let you specify how many shots the camera should fire and the time interval between them. For example, the command below takes five photos at three-second intervals:

```
gphoto2 --capture-image --hook-script /path/to/rsync_backup.sh
```

This can come in handy for time-lapse photography. Another useful parameter is `--hook-script`. As the name suggests, it allows you to attach a script to the `gphoto2` command. For example, you can create a simple script that backs up photos in a specific folder with the use of the `rsync` tool and then attach the script to the `gphoto2 --get-all-files` command:

```
gphoto2 --get-all-files --hook-script /path/to/rsync_backup.sh
```

This way, you can download photos from your camera and back them up in one go.

If using the command-line tool to download photos from the camera is not your cup of tea, use `gtkam`, a simple graphical front end to `gPhoto2` (Figure 2), to establish a connection to your camera that lets you browse, view, and download photos.

To connect `gtkam` to your camera, choose *Camera* | *Add camera*, select the appropriate camera model and port, and press *OK*. Then press the *Rescan* button to connect to the camera and fetch thumbnails. Now you can browse the photos on your camera. To view a specific photo, right-click on its thumbnail and choose *View with | Built-in viewer* from the context menu. The same menu also offers the *Save as* command, which allows you to download the selected photo to the netbook.

Phatch

Although your netbook doesn't have enough power and screen real estate for serious image editing, it can cope with light tweaking, like adjusting contrast and brightness or resizing, watermarking, and rotating photos, for example. To do all this on your netbook, you need `Phatch` [2] – an easy-to-use graphical batch processing tool (Figure 3). Unlike traditional image editing applications, `Phatch` doesn't allow you to edit photos directly. Instead, you create actions. An

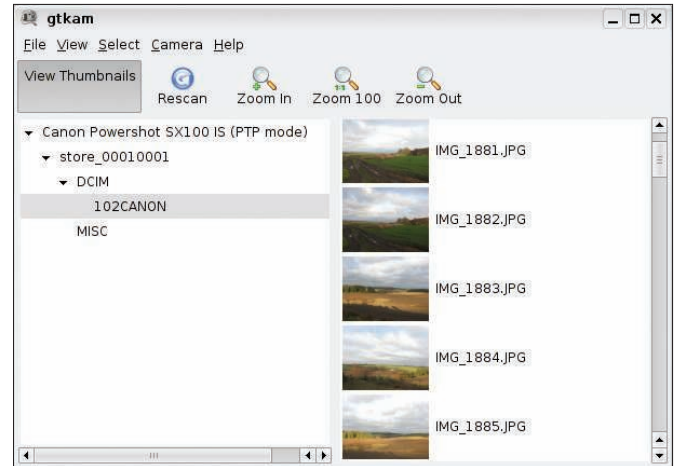


Figure 2: Gtkam is a simple graphical tool for viewing and downloading photos from your camera.

action is a single operation that the application performs on the photos that you feed to it. Each action provides a number of options; for example, the *Auto Contrast* action has two options, *Cutoff* and *Amount*, and you can specify as many actions as you like and save them as a single action list. To add an action to `Phatch`, press the *Add Action* button and select the action you want from the list. By selecting a category from the *Select* drop-down list, you can narrow the list to a specific action type. Alternatively, you can use the *Search* feature to search for a particular action.

Say you want to add an action that applies a text watermark to your photos. Choose the *Text* action and press the

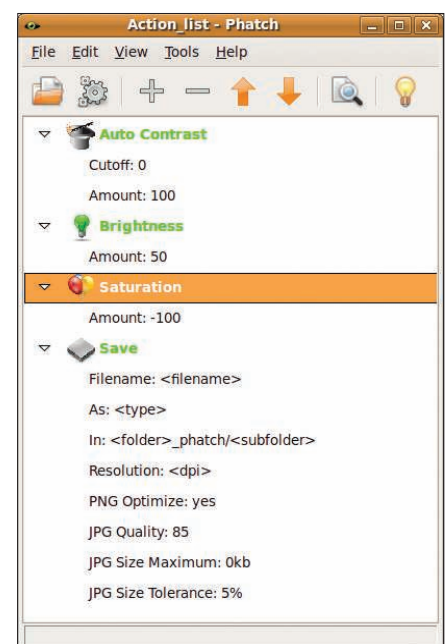


Figure 3: Phatch lets you process your photos in batches.

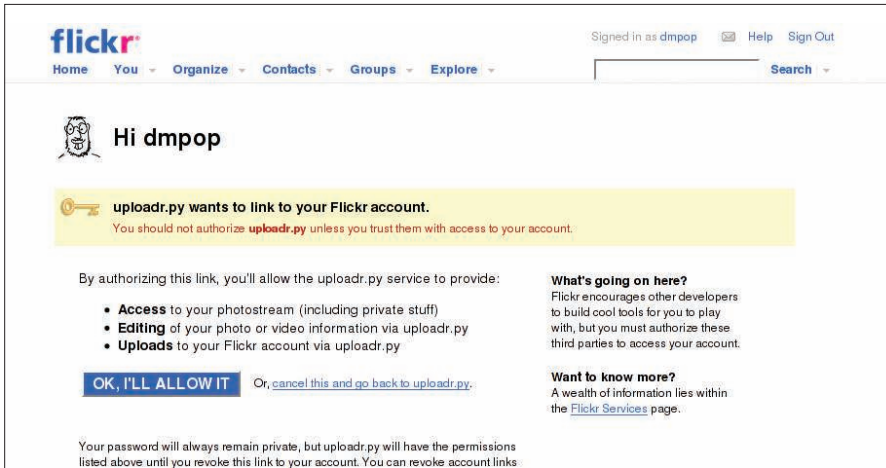


Figure 4: Before you can use the uploadr.py script, you have to authorize it.

Add button. Then use the available options to tweak the action. For example, the *Text* option allows you to specify the text of the watermark. With the use of variables, you can either specify a static text (e.g., “Copyright (c) Dmitri Popov, 2009”) or create a dynamic watermark.

If you want the watermark to include a timestamp, you can use the appropriate variables such as `<day>`, `<hour>`, `<minute>`, and `<second>`:

```
Copyright (c) Dmitri Popov, 2
<year> Time: <month>/<day> 2
<hour>:<minute>:<second>
```

Using other available options, you can specify orientation, font, size, and other watermark properties. Once you are satisfied with the settings, you have to add the mandatory *Save* action; without it, Phatch won't know where to save the processed photos.

Besides options such as output format and quality, the *Save* action also offers a set of variables that you can use to specify a destination for the processed photos. For example, the default `folder_phatch/subfolder` variable tells Phatch to save the processed images in a separate folder inside the source directory, whereas the `root/phatch/year/month/day` variable creates a directory structure on the basis of the current date. Except for the *Save* action, which must always be the last action in the list, you can rearrange the actions in the list by drag and drop.

To process photos with the action list, press the *Execute* button, choose the directory containing the photos you want to process, and press the *Batch* button.

Besides the *Execute* button, Phatch offers another feature that makes the batch process even more straightforward. Choose *View | Droplet*, and Phatch becomes a floating icon. Drag photos onto it, and the application processes them with the currently opened action list.

Uploading Photos to Flickr

Flickr is not only the best service for sharing photos with the world – it also makes a great backup solution. Dozens of utilities are available for you to use to upload photos to your Flickr account. But when you are on the move, you probably would want the most simple and lightweight tool for the job. And the `uploadr.py` [3] Python script meets both requirements.

To make the script work on your netbook, download the `uploadr.txt` and `xmltramp.txt` files and rename them `uploadr.py` and `xmltramp.py`. Now move the `xmltramp.py` file to the `/usr/lib/python2.5/site-packages` directory (if needed, replace `python2.5` with the Python version installed on your system).

Next, you have to configure the settings in the `uploadr.py` script. To do this, open the script in a text editor, locate the `IMAGE_DIR = "images/"` line, and replace `"images/"` with the path to the directory containing the photos.

Use the *Flickr* section of the script to specify settings for uploaded photos, for example:

```
Flickr = {"title": "",
         "description": 2
         "Uploaded from my netbook",
         "tags": "backup",
         "is_public": "0",
```

```
"is_friend": "0",
"is_family": "0" }
```

Now run the `uploadr.py` script with the following command:

```
python uploadr.py
```

Next, authorize the script to access your Flickr account (Figure 4). When you launch the script, it checks the specified directory for new pictures and automatically uploads them. By default, the script scans the directory every minute, but you can change that by modifying the `SLEEP_TIME = 1 * 60` line in the `uploadr.py` file. Also, you can run the script as daemon with the `-d` switch:

```
python uploadr.py -d
```

If you don't like to work from the command line, there are plenty of graphical utilities that let you upload your photos to Flickr. FlickrUploadr [4], for example, wraps the `uploadr.py` script into a simple graphical interface. To install FlickrUploadr, download the latest archive, unpack it, switch to the resulting directory in the terminal, and run the following command as root:

```
python setup.py install
```

Now you can start FlickrUploadr by running the `Uploadr` command in the terminal. To upload photos to Flickr, drag them onto the Uploadr window, fill out the fields in the opened Pictures information window, and press *OK*.

Final Word

A netbook won't replace your desktop machine for processing and managing your digital photos. But with the right software, a netbook makes a perfect digital photography companion that you can use to load photos off your camera, perform light tweaks on them, and upload them to Flickr. ■

INFO

[1] gPhoto2: <http://www.gphoto.org>

[2] Phatch: photobatch.stani.be

[3] uploadr.py script: berserk.org/uploadr/

[4] FlickrUploadr: micampe.it/projects/flickruploadr