

jigl

In Chapter 9 of **Linux for Non-Geeks**, you worked with a Python script called pyWings. Now you'll have the chance to work with a very cool and useful script written in another programming language, Perl. The script, which I first saw mentioned in the *Linux Journal*, is called **jigl** (Jason's Image Gallery). The script creates an html photo gallery (an example of which you can see in Figure 1) from any folder in which you keep .jpg, .gif, or .png images. Oh, and in case you are worried about such things, jigl is easy to install and, despite despite being command driven, very, very, very easy to use.



Figure 1: An html image gallery created with jigl

Getting the Files You Need for [jigl](#)

To install [jigl](#), you will need two or three files, depending on what you've got installed: ImageMagick, the [jigl](#) tarball itself, and the [jhead](#) package upon which [jigl](#) depends. First, check to see whether or not you have ImageMagick installed. To do this, open a Terminal window and type in `rpm -q ImageMagick`, and then hit `ENTER`. If you have it installed, RPM will tell you what version you've got; if not, RPM will also tell you so.

If ImageMagick is not installed on your system, the easiest way to go about things is to download and install it via APT/Synaptic, which you learned to use in Chapter 10. Once you have done that, you are ready to move on.

Next, go to <http://xome.net/projects/jigl/download.html> and scroll down to the Download section to get the Linux version of [jigl](#) tarball, the link for which is [jigl-2.0.1.tar.gz – Version 2.0.1](#) (at the time of this writing). Once you've done that, scroll up to the [jhead](#) link and click it to get to the [jhead](#) download page. You can just type in the URL, if you prefer that: <http://www.sentex.net/~7Emwandel/jhead/>.

Once on the [jhead](#) page, scroll way down to the list of available download files and select [jhead-2.1-0.i386.rpm](#), which is right next to the words **Linux binary RPM (RedHat 9.0)**, which works just fine with Fedora Core.

Installing [jhead](#) and [jigl](#)

OK, so now that you have downloaded, and they are sitting pretty (relatively speaking) on your hard disk, let's get down to installing them. You will start with the dependency, [jhead](#), first.

1. Install the [jhead](#) RPM package by double-clicking it and following the usual RPM installation procedures. See Chapter 8, if you've forgotten.
2. Untar the [Jigl](#) tarball by double-clicking it and then going through the usual File Roller

procedures. See Chapter 9, if you've forgotten.

3. Open a new Terminal window and become root using the *su* command.
4. Now *cd* over to the new folder created by the *jigl* tarball. As of this writing, that would be *jigl-2.0.1*, so just type in `cd jig1-2.0.1` and hit ENTER.
5. Now you are going to copy the **jigl.pl** file into your system's command search path in root territory, so that everyone can use it easily. In your Terminal, type in `cp jig1.pl /usr/local/bin` and then press ENTER.
6. Now you have to change the permissions of the *jigl.pl* script in order to make it executable, after which you will be able to run it like a command. In your Terminal type in `chmod a+rx /usr/local/bin/jigl.pl` and hit ENTER.
7. OK, so now that you are done with the installation process, you can dump the *jigl-2.0.1* folder. You can also dump the *jigl* tarball and *jhead* RPM, though I would back up those two files to CD first, just in case you need them again in the future.

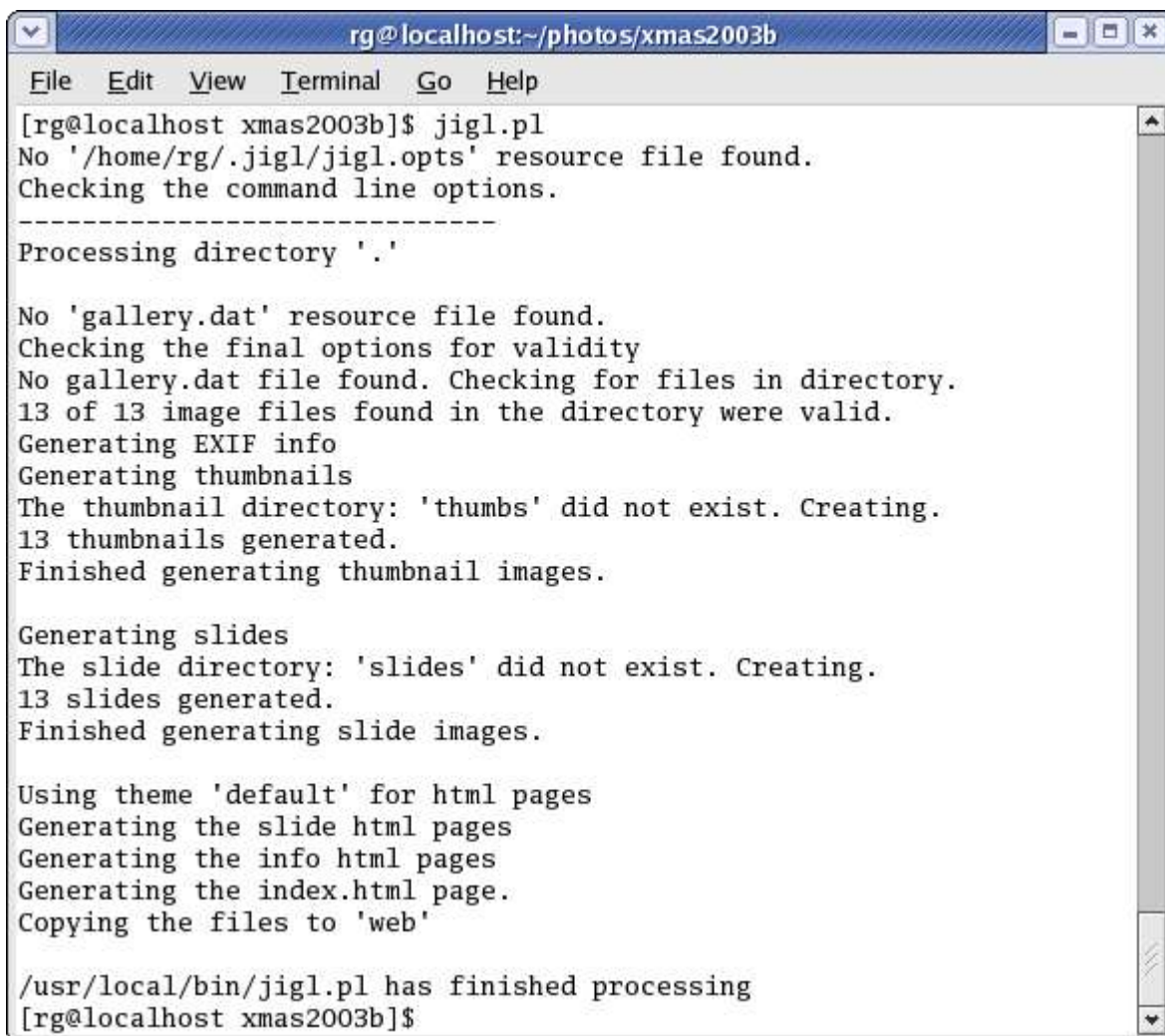
Using **jigl**

Although there is no graphical interface for *Jigl*, it is surprisingly easy to use. Just open a Terminal window and *cd* over to the folder where you keep your images. For example, if you keep your photos in the *photos* folder I asked you to create in Chapter 5, then you would type `cd photos/` followed by ENTER. After that, all you have to do is type in `jigl.pl` and press ENTER. The magic will then begin.

The progress of this “magic” will be shown in the Terminal window (Figure 2). First, *jigl* will examine what options you selected (more on that later), and then it will inspect the contents of the target folder (*photos/xmas2003b/* in the example) to see if all is hunky dory. Once *jigl* has decided that it can deal with what you've thrown at it, it begins its work. First it gathers all the information that your

digital camera stores with each photo it takes (Exif files). Then it creates thumbnail images for each of the images in the target folder and a folder in which to place them (thumbs). These are used for the gallery index page. Next, jig1 goes on to create slides of each image (these are the images that will be actually be shown in the individual image pages) and a folder (slides) in which to place these as well.

Finally, jig1 creates the html pages: one slide page for each image, one info page for each image, and one index page for the whole shebang. These are placed in a folder that jig1 creates specially for this purpose – called **web**.



```
rg@localhost:~/photos/xmas2003b
File Edit View Terminal Go Help
[rg@localhost xmas2003b]$ jig1.pl
No '/home/rg/.jigl/jigl.opts' resource file found.
Checking the command line options.
-----
Processing directory '.'

No 'gallery.dat' resource file found.
Checking the final options for validity
No gallery.dat file found. Checking for files in directory.
13 of 13 image files found in the directory were valid.
Generating EXIF info
Generating thumbnails
The thumbnail directory: 'thumbs' did not exist. Creating.
13 thumbnails generated.
Finished generating thumbnail images.

Generating slides
The slide directory: 'slides' did not exist. Creating.
13 slides generated.
Finished generating slide images.

Using theme 'default' for html pages
Generating the slide html pages
Generating the info html pages
Generating the index.html page.
Copying the files to 'web'

/usr/local/bin/jigl.pl has finished processing
[rg@localhost xmas2003b]$
```

Figure 2: jig1 shows its progress in the command Terminal

Viewing the Results

Once `jigl` is done doing its thing, you can exit out of the Terminal. After that, open a Nautilus window and then open up the folder where you are keeping the images you just `jigl`-ized. Inside that folder, you will find the new folders mentioned in the previous section (`thumbs`, `slides`, `web`). Double-click the one called `web`, and then look for the file `index.html` inside. Double-clicking `index.html` will cause it to open up in your Mozilla web browser. The output should look more or less like that in Figure 1, albeit with your images, not mine.

To see a slide of one of the thumbnail images, just click the image you want a closer look at. It will then open up in its own page, an example of which you can see in Figure 3.

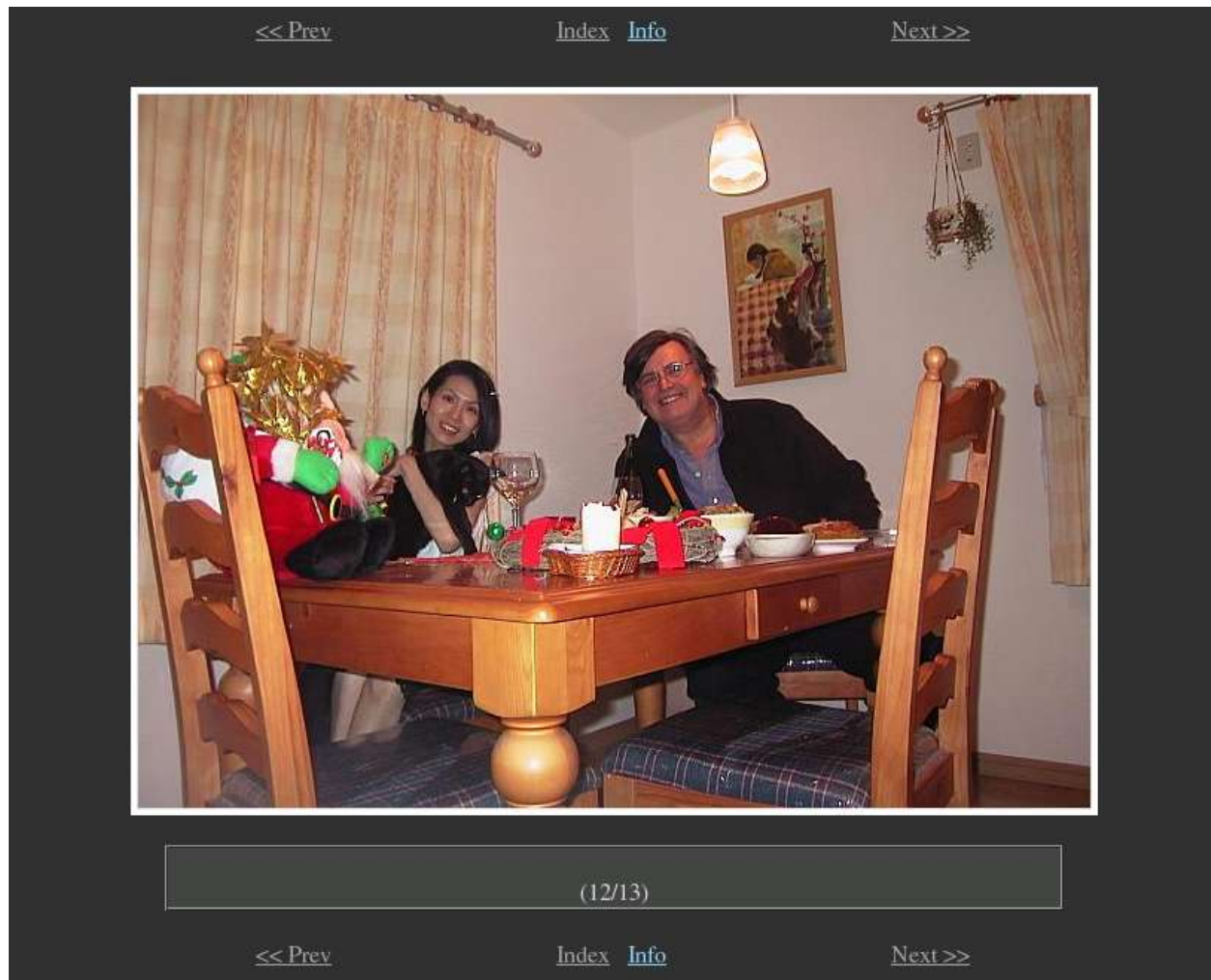


Figure 3: A slide html page created by `jigl`

From the slide page, you have a number of navigational options open to you. You can simply return to the index page by clicking the [Index](#) link, move backwards and forwards amongst the various slides by using the [Prev](#) and [Next](#) links. If you're into photography, as the creator of this script seems to have been, you will be especially pleased when you click the [Info](#) link. As you can see in Figure 4, the Info page for each image contains all the technical details of that image: shutter speed, aperture, metering type, resolution, focal length (useful when using a zoom), focus distance, and even the make and model of the camera. Very cool.

File name	: tim_3020.jpg
File size	: 320544 bytes
File date	: 2003:12:25 18:52:12
Camera make	: Canon
Camera model	: Canon PowerShot A50
Date/Time	: 2003:12:25 18:52:12
Resolution	: 1280 x 960
Flash used	: Yes
Focal length	: 4.3mm (35mm equivalent: 29mm)
CCD width	: 5.31mm
Exposure time	: 0.033 s (1/30)
Aperture	: f/2.6
Focus dist.	: 32.77m
Metering Mode	: center weight
Jpeg process	: Baseline

Figure 4: An info html page created by [jigl](#)

Options

There are a number of options available to you with [jigl](#). If you want have a look at all of them, just

<http://www.edgy-penguins.org/non-geeks/>

open a Terminal window, type in `jigl.pl --help` (notice that there are two hyphens before the word `help`), and then hit `ENTER`. You can also find a lot of information at <http://xome.net/projects/jigl/usage.html>. For my part, I am just going to mention a one of these choices, which you should also give you a pretty good idea of how to use the others.

Watermarks

My favorite option is adding watermarks. Basically this is done by adding the flag `-ws` (watermark slides) in conjunction with `-wf` (watermark file). Give it a try by putting a nice red apple in the corner of each of your gallery slides. Oh, and don't worry, your original images will not be touched. Here is what you have to do:

1. As this is just experiment, let's use a folder with only a few images – 10 would be about right.
2. Open a terminal and `cd` over to that folder. Let's say the folder is called *weddingpics*, which is located in your *photos* folder. In this case, you would type:

```
cd photos/weddingpics/
```

 (and then tap `ENTER`).
3. OK, now you will use the *jigl.pl* command followed by the two flags, `-ws` (to tell *jigl* to place watermarks on the slides, and `-wf`, followed by the location of the watermark image (to tell *jigl* what image you want to use as a watermark). In this case, we are going to use the image called `apple-red.png` which is located in `/usr/share/pixmaps`. This being the case, you would thus type in the following line, `jigl.pl -ws -wf /usr/share/pixmaps/apple-red.png`, followed by a press on the `ENTER` key.



Figure 5: *jigl* places a watermark in your image

Rickford Grant - **Linux for Non-Geeks** Supplementary Projects: **jigl** 8

As you can see in Figure 5 above, the little red apple icon is now shown in the bottom right corner of the image. Pretty cool, eh?

Now you are ready to move one and play with `jigl` yourself. Remember, to find out more on how to use `jigl`, just go open a Terminal window and type in `jigl.pl --help` (notice that there are two hyphens before the word `help`), and then hit `ENTER`. And don't forget that there is also a lot of information at <http://xome.net/projects/jigl/usage.html>.

Have fun!