

Using JPG Comments for Efficient Image Processing

John Nash

June 24, 2018, revised March 9, 2025

Motivation

The now-common JPG or JPEG files that are often the file format for photographs from digital cameras and cell-phones can store large amounts of text in several fields. In particular the **Comment** field is relatively accessible. Furthermore, having the “caption” within the file avoids loss of this information when the name of the file is changed. Some photo-management software separates the caption and image, and thereby risks caption loss.

This document discusses how to leverage the use of these comments in processing collections of images into article and book length objects. In particular, fairly simple file naming and organizing practices lead to highly efficient workflows.

Unfortunately, the majority of image display programs do not, by default, display the Comment field. Below I discuss some different tools that can be used for display and edit of the Comments.

A selection of tools

Mapivi

In the past, we have used the program **mapivi** which is nominally available for multiple platforms. However, while I have found it easy to install for Linux, particularly distributions in the Debian family, I have not found the Windows executable to install satisfactorily. The more recent Perl code (https://sourceforge.net/projects/mapivi/files/Mapivi_Pperl/) suggests a cross-platform possibility, but my experience is not encouraging.

Irfanview

The program Irfanview is a well-established and feature-rich image viewing and management program. (<http://www.irfanview.com/>) It is free for non-commercial use. Moreover, while it is primarily for the Windows platform, the 32 bit version can be installed on Linux using WINE (a WINdows Emulator). Irfanview allows the Comment field to be added or edited by using the menu entry Image / Information / Comment. Moreover, there are viewing options that allow one to cycle through images in a directory with the Comment displayed. However, I have found that installation and use can be prone to glitches in Linux.

WARNING: I have found download sites for Irfanview are designed so that the most obvious “Download” button will try to install software that has nothing to do with Irfanview and which is – if not outright malware – generally deleterious to your system. Make sure you are actually downloading the Irfanview program and/or its plugins and nothing else. Note that for installation under Wine you will likely need the 32-bit version.

feh

This Linux image viewer is fast and lightweight, but with some cleverness offers a rather nice caption display. Information may be found at <https://github.com/derf/feh> or <https://feh.finalrewind.org/>.

The caption display uses another program ‘rdjpgcom’ to get the comment/caption. This is packaged in my Bash script ‘imgcmtx’ and invoked with

```
imgcmtx path_to_directory
```

Text of the script:

```
#!/bin/bash
# imgcmtx -- display and browse images with comments
echo "Display image $1 and then images in same directory."
#
# echo "PWD variable is $PWD"
HERE=`dirname $(realpath $1)` # must execute
echo "Current directory is $HERE"
# set then reset font
/usr/share/fonts/truetype/freefont/FreeMono/22 $1 $HERE/
feh --info 'echo %f; rdjpgcom %F' -F -e
/usr/share/fonts/truetype/freefont/FreeMono/14 $1 $HERE/
```

JNJpgCmt.py

The main motivation for my writing this program was that we needed to efficiently check and edit JPG comments, especially for genealogical images which often have quite extensive comments. A secondary goal was to learn to program in Python. The program is functional, though some keyboard shortcuts have not yet been incorporated. Assistance is welcome to improve the code, as I have not found documentation that allows me to make the required changes. The program uses subsidiary programs ‘rdjpgcom’ and ‘wrjpgcom’ to read and write the comment blocks from and to the jpg file.

The file ‘JNJpgCom-settings.xls’ is a workfile to collect notes on the settings that appear to be suitable for my computers when running this program. That is, to get the sizes of windows and images appropriate to my monitors and the images I work with.

jpgautorot

When taking pictures with electronic cameras and cell-phones, the user will often rotate the device to get an appropriate landscape or portrait oriented image. However, the stored image may be in an undesired orientation. Many devices add an EXIF tag based on camera sensors to indicate the “up” direction for the image. This Bash script first renames all files with extension “.JPG” or “.jpeg” to “.jpg” then uses the program **jhead** to reorient the image file. This script should be run in a terminal opened in the appropriate file directory.

jpgdir2htm.pl

This program is intended to allow for distribution of captioned images so that they can be viewed and downloaded by others. It is the latest in a line of programs to produce HTML presentations of captioned photo files.

This Perl script finds all *.jpg files (NOTE only lower case in the 20250311 version) then extracts the comments and arranges a commented display in an HTML file. The jpg files are displayed in alphabetically sorted order. It also builds
build a zip file with output HTML files and all images ready for upload to a web site, and finally calls the program **gftp** to allow the upload to an appropriate server.

jpgcom2txt

This Bash script generates a sorted list of files with their embedded captions. The result is a file ‘out.txt’ with the content.

Long captions are split to new lines that are prefaced by a space.

At 2025-3-14 the script does NOT accept a path as first argument, so needs to be launched from within the appropriate directory.

pix2fotohtml.pl

This is a Perl script that is an enhanced version of **jpgcom2txt**. See the section “Presentations of photos” below.

jpgsindir2pdf

This Bash script assumes a first argument that is a directory name that is in the current active path. (This can be provided by an active icon in Double Commander, and we have done this on our own computers, where highlighting a directory of images and clicking he icon runs the script.)

The script creates a list of the files of type “.jpg” in the directory (note that lower case is used for the file extension, so files MUST be renamed first if they are not of this form). The file ordering is whatever the current “glob” rules use. The files in the list are then converted to pdf files using ImageMagick’s **convert** program, and then concatenated using **pdftk** to a single pdf file that is given the name of the directory with “.pdf” appended, which can then be adjusted using the program **pdfarranger** that is launched from the script. The script then offers to delete all the intermediate pdf files.

Note: the consolidated file is placed at the same path level as the target directory of the script. Do NOT invoke this in a bottom-level directory where the jpg files are located or you will get incorrect results.

fotohtml.pl and its antecedents

This program is launched with the command

```
fotohtml.pl myfile.txt
```

where ‘myfile.txt’ is an appropriately structured list of image filenames and captions. It processes this list into a set of linked HTML files that display the images with captions. Each file has a maximum of images (default 10), with the number controlled by the variable ‘\$fperpage’ in the Perl script.

Editing Comments with JNJpgCom.py

Launching the program within a directory with images will display one of the images with its caption (if any) in a pink box below the image. There are buttons to navigate to the next or previous image, to Exit, to get Help, to Save the comment (if edited) and to Rename the file.

Specifying the image filename on launch will display that image first.

Editing Comments with Irfanview

To add/edit comments with Irfanview, one first opens an image using the file selector (either File/Open or click on the small folder icon), selects the desired file, and chooses Open.

The Comment can be viewed and edited by choosing (Image / Information / Comment), or using the keyboard shortcut I followed by Alt-C. If the Comment is changed, then one must Write it, and then File / Save (shortcut Ctrl-S) to ensure it is back on disk.

Configuration of Irfanview for Comment editing and viewing

The full-screen view of Irfanview can be made to display the Comment as well as other data. The options can be set by View / “Show fullscreen options” or use the keyboard shortcut P and select “Fullscreen / Slideshow”. Note that finding the documentation of these possibilities may require some effort.

I have found that it is helpful to set the following options:

- Under “Set display multiplier” (upper right of option box) I like 80% for height and 90% for width.
- I like the text position “bottom” and I turn off transparent background. (On one machine we found it necessary to turn off transparent background to change text colour.)
- I like a black text background and screen color.
- I like to use Courier New 12 point and make the text yellow, noting that it may be necessary to play with other options before this is possible. I have also noted that some machines may not have Courier New available, in which case another font needs to be chosen.
- Put \$C as the only entry in the “Full screen only” text box, and make sure “Show text” is selected.

These choices put the Comment at the bottom and leave enough room that it generally does not overlap the image. They also make it a readable size and colour. However, I encourage users to play with the full-screen display options to accommodate their own needs and preferences.

To choose full-screen display we simply hit the Enter key. (Esc will exit full-screen.) The space bar cycles through the files in the currently-selected directory.

WARNING

Unfortunately, in Irfanview there is a setting in the JPEG/GIF Save dialog that **MUST** be set ON if one is not to lose the comments. This is “Keep original JPG-Comment”. Make sure it is ON when you Save or Save_As.

Accessing and managing files (Double Commander)

In our home/business operations, we have found a two-panel file manager of great help. Recently, we have been using Double Commander (<https://doublecommander.com/>). This program, written in Free Pascal by Alexander Koblov and others, is highly configurable and works quite well. We have found we can create our own “buttons” to carry out operations, in particular for syncing directories, clearing duplicates etc., rendering collections of images to pdf documents, and other tasks.

Presentations of photos

We have found over the last 20 years that HTML (World Wide Web) files are a good way to present images with associated text. Our progress with such files has been stepwise:

- static HTML files with text and embedded images
- the **fotohtml** program, now ‘fotohtml25.pl’ is a Perl program to take a collection of image files and a text file with captions and process them into a Web presentation, segmenting the “pages” so each has a maximum number of images (default 10)
- automation of the generation of the caption file with *pix2fotohtml*, which takes a directory containing JPG images and a file “@TITLE@.TXT” with a 1-line title and generates the caption file that, with the images, can be processed by **fotohtml**. If the title file is not provided, a title should be added into the caption file manually with a text editor.

The last stage is the most recent. We further refine the process by imposing some simple file naming conventions. These are intended to ease the task of keeping track of image files and their content by providing a natural chronological sort order.

- each file starts with a year ‘yyyy’ to which it may be ascribed
- if the year is inexact, we use ‘yyyc-’
- if the time frame is a decade, then we use ‘yyyys-’

Our program *pix2fotohtml* sorts files by name, hence by the year attributed. Clearly other sorting rules could be proposed. However, we have found this works quite well.