Showing JPG comments in an image browser

John C. Nash

2017-06-04

The Conclusions (up front!)

The JPEG image standard allows several ways to save caption information INSIDE an individual JPG file. This is important, as captions stored separately can be "lost" if filenames are changed. However, **which** choice of mechanism to use leads to confusion. Here I recommend the "Comment" field be used. This is available via **rdjpgcom** and **wrjpgcom** tools in Linux. On Windows, or under WINE in Linux, Irfanview is quite capable (Image/Info/Comment), but **mapivi** which is cross platform is sometimes better for adding comments. For Linux I found **feh** allowed a simple script to display images with comments for a quick verification of the caption. Irfanview offers a full-screen option that is similar.

Having got into "production" mode, I discovered that it was, or at least feels, more efficient to work mostly in Irfanview. The background to this is outlined in the section "Production workflow".

Motivations

My wife and I have a lot of photos. We don't look at them very often, but they do represent memories and serve to remind us of events in our past. There are also some that should be passed to other family and friends. The many albums take up space, and are not terribly well stored. For this reason we have been scanning them.

Images on their own are not particularly useful. Therefore we have been adding captions, using the "Comment" capability of JPG files. This is in the IPTC tags of the JPEG standard (https://en.wikipedia.org/wiki/JPEG).

We have found the program **mapivi** (http://mapivi.sourceforge.net/mapivi.shtml) to be one of the best tools for adding comments to a JGP image. Unfortunately, it is not a very friendly program for simply browsing the images quickly. We would prefer something like **eog** (Eye of Gnome Image Viewer – https://help.gnome.org/users/eog/stable/) or **xviewer** (https://github.com/linuxmint/xviewer) which is the default in the Linux Mint 18 operating system I use. Version 1.0.6 is installed. For quite serious examination and adjustment of images, I use GIMP, the Gnu Image Manipulation Program (https://www.gimp.org/).

Other image programs I have considered – but not necessarily tested – are:

- pix (https://github.com/linuxmint/pix)
- gthumb (https://sourceforge.net/projects/gthumb/)
- gwenview (http://gwenview.sourceforge.net/overview)
- Irfanview (a Windows program run under WINE) http://www.irfanview.com/
- geeqie (http://geeqie.sourceforge.net/)
- shotwell (https://en.wikipedia.org/wiki/Shotwell_(software))
- imagemagick `display` (https://www.imagemagick.org/script/index.php)
- feh (https://feh.finalrewind.org/)
- digikam (https://www.digikam.org/)

Further complicating matters is the fact that while several programs offer the capability of viewing and editing comments, they do not seem to show the comments added by mapivi. It was this last problem that prompted this investigation and request for help on the OCLUG linux list.

Test Sequence

The following is my workflow, illustrated with a single photo taken some years ago (1992-9-8 to be precise).

Scanning

I scan photos using a flatbed scanner driven by a small **bash** script called **scanj17**:

```
#!/bin/bash
# Copyright (C) John C Nash 2009
## To scan multiple pages and increment the names of the image files
##
## May not perfectly correct images
## needs imagemagick (possibly other packages)
echo "SCANJ17   J C Nash 2017 -- scan multiple pages to incremented JPG files"
# echo "You may need to fix the scanner string 'scanimage -L' to detect"

echo ""
echo "Files currently present"
ls
echo ""
echo "Enter last index of scan (0 to start anew)"
read ii
echo "Starting at $ii"

END_CONDITION='X'

echo "Put document page on scanner; Enter to continue, X to stop"
read var1

until [ "$var1" = "$END_CONDITION" ]
# Tests condition here, at top of loop.
do
  let ii=ii+1
  jj=$(printf %03d $ii)
  echo "scanning page  $jj"

  scanimage --resolution 300 --mode color >image.pnm
  convert -quality 95 image.pnm imagex.jpg
  cp imagex.jpg "scanimg$jj.jpg"

  echo "Put document page on scanner; Enter to continue, X to stop"
  read var1
#  echo "variable #1 = $var1"
#  echo
done

echo "Rename images - no entry skips renaming"
read -p newname: newname
if [ "$newname" != "" ]; then
    echo "RENAMING scanimg??.jpg to $newname??.jpg"
    rename s/scanimg/$newname/ *.jpg
fi
```

```
echo "Files:"
ls
echo "DONE!"
# Srm image.pnm
rm imagex.jpg
```

We use this to create the file `scanj17-raw001.jpg` by entering the name `scanj17-raw` after stopping the scan cycle. Note that we have suppressed deletion of the intermediate pnm file.

Properties of the raw scan

Using GIMP 2.8.16, we find the `image.pnm` file produced by **scanimage** to be 2574×3531 pixels. It is 27,266,419 bytes in length.

Converting this with the **imagemagick convert** utility results in the `scanj17-raw001.jpg` file which has the same size in pixels but is only 2,162,535 bytes. (We use an image quality setting of 95 in the conversion.) GIMP shows no comment for the pnm file, but the jpg file has the comment

```
 SANE data follows
```

This seems to have been added by `convert` somehow. I believe its presence may be causing some of the troubles I have had in seeing comments added by **mapivi**.

Cropping the raw scan

The raw scan is of the entire scanner bed. Therefore there is much whitespace around the image we wanted to capture. First we copy the file to `cropped.jpg`, then used **gwenview** to do the cropping. GIMP now shows a file that has 1054 by 1465 pixels. It uses 146,618 bytes. The comment is still "SANE data follows" as shown by GIMP. **rdjpgcom** shows the comment to be

```
SANE data follows
\000
```

Adding a comment with mapivi

Opening `cropped.jpg` in **mapivi**, I allowed the thumbnails to be generated for all jpg files, then selected the `cropped.jpg` file. Note that this behaviour of trying to examine all files in the directory is a nuisance.

I then tried using the `a` command and added "Mendocino: pumpkin patch". I then opened another file, then returned to `cropped.jpg` as **mapivi** does not display the comment immediately. In this case, it did not display any comment except the "SANE data follows". I thought perhaps I had failed to click "OK" after adding the comment and repeated my efforts. Still no joy. Then I tried **rdjpgcom** and was rewarded with the comment:

SANE data follows \000 Mendocino. Pumpkin patch. Mendocino: Pumpkin patch.

(Note my typing was not exactly the same in each case.) I found that the original un-cropped file did NOT have the "\000" element. GIMP only shows the "SANE data follows". However, **mapivi**, when I try the edit command `e`, asks me which comment to edit. I chose the "SANE data follows" and erased all content. Then the second element "Mendocino. Pumpkin patch." After OK, I then tried to edit the comment, but was asked "Please select one of three comments to edit". Selecting the only one which showed, I was offered a blank box! However, GIMP now shows "Mendocino: Pumpkin patch." preceded by 2 blank lines.

It seems that there is a bug in several pieces of software giving inconsistent behaviour.

I was able to "fix" the issue by opening the `cropped.jpg` file in **mapivi** by right clicking on it and selecting "join", then choosing "newline" as the separator. This put all the comments into one, which then displayed appropriately.

A cleaning script

The following is a recipe to fix files with comment blocks like this. We will start with myfile.jpg that has multiple comment blocks and possible "\000" entries.

- Copy the comments to a text file:

  rdjpgcom myfile.jpg > tt.txt

- Clean up the comments with a text editor or a script.

  In a manual edit we remove any special entries like "\000"

  A manual edit permits fixing other errors in the comment(s). We could also remove the "SANE data follows" entry as well.

  Save the file, either as tt.txt or another name.

- Change the name of the file.

  mv cropped.jpg ctemp.jpg

- Put the comments back using **wrjpgcom

  wrjpgcom -replace -cfile tt.txt ctemp.jpg >cropped.jpg

  The `-replace` tag is critical to removing troublesome comment material.

This process was successful, and the resulting script `fixjpgcom` follows:

```
#!/bin/bash
# fixjpgcom -- a script to clean up comments in jpg files
rm ttt
for myfile in $1/*.jpg
do
   echo "Processing $myfile"
   rdjpgcom $myfile > ttt
   echo "Original caption:"
   cat ttt
   sed 's/ SANE data follows//' ttt > xxx
   sed '/^$/d' xxx >ttt
   echo "About to write back caption:"
   cat ttt
   echo "----------------------"
   wrjpgcom -replace -cfile ttt $myfile > out.jpg
   echo "moving out.jpg to $myfile"
   mv out.jpg $myfile
   rm ttt
   rm xxx
   echo "done $myfile"
done;
```

Confusions about comments

In the process of trying to understand the difficulties I was having, I examined a number of software packages as listed above. In particular, it seemed fairly easy to use **pix** to add or edit a comment. However, I could never find the comments I had added with **mapivi** or **wrjpgcom** to my jpg images. I then created a file `ccpix.jpg` and used

mv ccpix.jpg temp.jpg wrjpgcom -replace -comment "" temp.jpg >ccpix.jpg rm temp.jpg

so that `ccpix.jpg` had no comment. I then used **pix** to add a comment

Comment added with pix

I then opened the file with Imagemagick's **display** and right clicked, then selected "Image Info". This showed that the comment was in the EXIF UserComment field. However, **mapivi** and **wrjpgcom** put the information in the (IPTC) field Comments, as does **GIMP** and **Irfanview** and . . .

Another tool is `iptc` from **libiptcdata-bin**. However,

iptc cc1a.jpg

showed no IPTC data. Thus the comments are not in this metadata.

ExifTool and pyExifToolGui

170423 – installed these. And I can show the "Comment", but trying to edit gives a popup message that IPTC is an "older" metadata standard and that XMP is the newer one. Also that features for IPTC won't be added. Sigh.

Note that installing the exiftool GUI required several dependencies relating to pyside to be installed. However, seems to work OK.

However, `exiftool` is quite capable, even if its use is daunting to a newcomer.

Some tasks where a recipe could be useful are as follows:

- getting UserComment
- getting Comment (IPTC?)
- understanding tagFromFile

A home-grown solution

Using Perl Tk

I attempted to modify programs in Perl Tk tutorials to display an image and comment (caption). It proved fairly easy to do parts of the task, but combining all the aspects of what I sought proved elusive.

By April 27, 2017, various programs were "sort of" working. The biggest difficulty I had with Tk is that it is difficult to keep the index of the current image (generated in one subroutine) to pass to another subroutine. The mainloop() structure also seemed to trigger events if the scaling of the image pushed the buttons off the main window.

Using an existing image program

I looked at a number of programs (see above). Then I found a note concerning the program **feh** that pointed out the `--info` tag could be followed by a command line program that returned text to the standard output.

Initially I had not chosen **feh** because it did not seem to scale the image using the `-size` option. However, I found that it did scale using the `-geometry` tag.

I combined these options and created a small script to record the choices I made. I call this **imgcmt**

```
#!/bin/bash
#  imgcmt -- browse images with comments
feh -g 800x1000 --draw-tinted --info "rdjpgcom %F"
```

This program allows easy cycling forwards or backwards through the imgage in the current directory using the left and right arrows. Comments are displayed at the bottom left. The image file name is at the top of the image window.

In production use, I found that a minor annoyance of this script was that it always started at the "top" of the file list (I believe in directory order, which is likely set in a configuration somewhere). It would be nice to be able to specify a particular file as the "starting" image. The documentation of feh suggested a possibility. See https://www.lifewire.com/feh-command-line-image-viewer-4054068. I created the script imgcmt1 and found that I could highlight a jpg file in Double Commander and "open with" this script once it was in my current path.

```
#!/usr/bin/perl -w
#  imgcmt1.pl -- display images with comments

use strict;

my $current_file = @ARGV[0];

system("feh -g 800x1000 --draw-tinted --info 'rdjpgcom %F'  $current_file");
```

Late June 2017 I found that there is a full-screen display option in Irfanview that allows the comment to be displayed with the image. Because this is useful for Windows users, in particular some family members with whom we wist to share old photos, I have separately documented this possibility. See ??

Customization of geeqie

I looked into the customization of geeqie, but could not get this program to display (or edit) the comment field. It seems to use the Exif UserComment field.

Production workflow

As we ramped up our effort to scan and caption many, many albums of photos, we discovered that we needed to adjust our workflow. Here is our current (2017-6-4) view.

- mapivi appears to have one or more glitches and occasionally locks up. Moreover, its requirement to create thumbnail images in each directory adds a set of nuisance files to our collection that we would rather avoid. Mapivi also requires the user to click on a file to display or redisplay the captions, which is a further nuisance. Nevertheless, the mapivi program is one of the better choices for captioning.
- The scanj17 script worked well on both our Canon LIDE 60 scanner and HP Deskjet 2130 All-in-one printer/scanner. However, images from the Canon were somewhat darker than those from the HP. This could be corrected in the XSane package by adjusting the Gamma control. An alternative turned out to be to run the Image/Auto_Adjust_colors in Irfanview. This could be done via a keyboard shortcut Shift-U. Similarly, Irfanview has some shortcuts for rotating and cropping which were very useful, since we aggregated photos into groups to scan multiple photos in one scan operation. We could add or edit comments by the two-step keyboard shortcut I (for Image), Alt-C (for Comment). Under Windows

operating systems, we have used Irfanview to actually operate the scanner, but this functionality does not seem to be readily available under Wine for Linux.

- We use the imgcmt script to review our comments/captions as a check that proper information has been saved before archiving our work.
- Within the archiving process, we found that small scripts to update a Synology backup server saved keystrokes and potential typing errors, especially if we had renamed directories, since we use rsync with the –delete tag active in some cases.

Conclusions

The JPEG image standard allows several ways to save caption information INSIDE an individual JPG file. This is important, as captions stored separately can be "lost" if filenames are changed. However, **which** choice of mechanism to use leads to confusion. Here I recommend the "Comment" field be used. This is available via **rdjpgcom** and **wrjpgcom** tools in Linux. On Windows, or under WINE in Linux, Irfanview is quite capable (Image/Info/Comment), but **mapivi** which is cross platform is sometimes better for adding comments. For Linux I found **feh** allowed a simple script to display images with comments for a quick verification of the caption. Irfanview offers a full-screen option that is similar.

Having got into "production" mode, I discovered that it was, or at least feels, more efficient to work mostly in Irfanview. The background to this is outlined in the section "Production workflow".