Part 13

Export data into files as metadata

Timelapse can export data by writing it directly into the image and (non-AVI) file as *embedded metadata*. This is the flip side of Part 9, which explained how we can read image metadata into Timelapse to ease data entry process, e.g., a file's date, specialized fields such as temperature, and others.

Should you write data into images?

Writing data as image metadata is useful only if you have a good reason, e.g.,

- Images and its stored metadata must stand alone. For example, if an image is mailed to someone, they can look up that metadata.
- If images are somehow separated from their management system, data can (potentially) be recovered from the image itself.
- You are relying on, or are migrating to/from, another system that is based on image metadata (this is rare).
- It plays a secondary role, where primary data storage and management is handled by other means (e.g., by Timeapse, other databases, csv files...).

Note. Storing data within images should not replace a primary data storage and management system. Using image metadata for image management, while possible, is not robust. It introduces performance issues, and is error-prone unless used very consistently. Formatting issues can occur (numbers vs. strings vs dates). Metadata may be altered by other software you are using. Specialized software (e.g., Exiftool) may be required to view or extract metadata. Centralized data management, efficient searching, global operations, and data recovery becomes challenging.

Background

Image metadata follows many standards that you may (or may not) have heard of. A single image can have different sets of metadata, each following its own standard. For example:

- *Exif*: describes fields containing technical camera information, such as shutter speed, focal length, date time, etc.
- *IPTC:* describes fields containing descriptive data, such as copyrights, captions, keyowrds, licensing and others.
- *XMP*: is more general purpose. It includes pre-defined fields (e.g., subject, star ratings, rights). More importantly, it also provides the ability to create custom metadata fields.

Image metadata can be thought of having three primary attributes:

- *Directory:* is a label that groups together related metadata. Think of it as similar to a folder or namespace, where metadata that share the directory name are considered as belonging together.
- *Tag ID:* is the name of the metadata property. It can also be called *tag, property, identifier, key* or *name.*
- *Value:* is the value associated with the TagID.

The directory also disambiguates tags that otherwise have the same name. For example, EXIF:Keywords is different from XMP:Keywords

Its actually more complicated than that, but this background is sufficient for Timelapse's purposes.

XMP-TimelapseData. Foreshadowing what is to come, Timelapse uses the XMP standard to export its data. That standard allows for custom directory names, and for custom tag names.

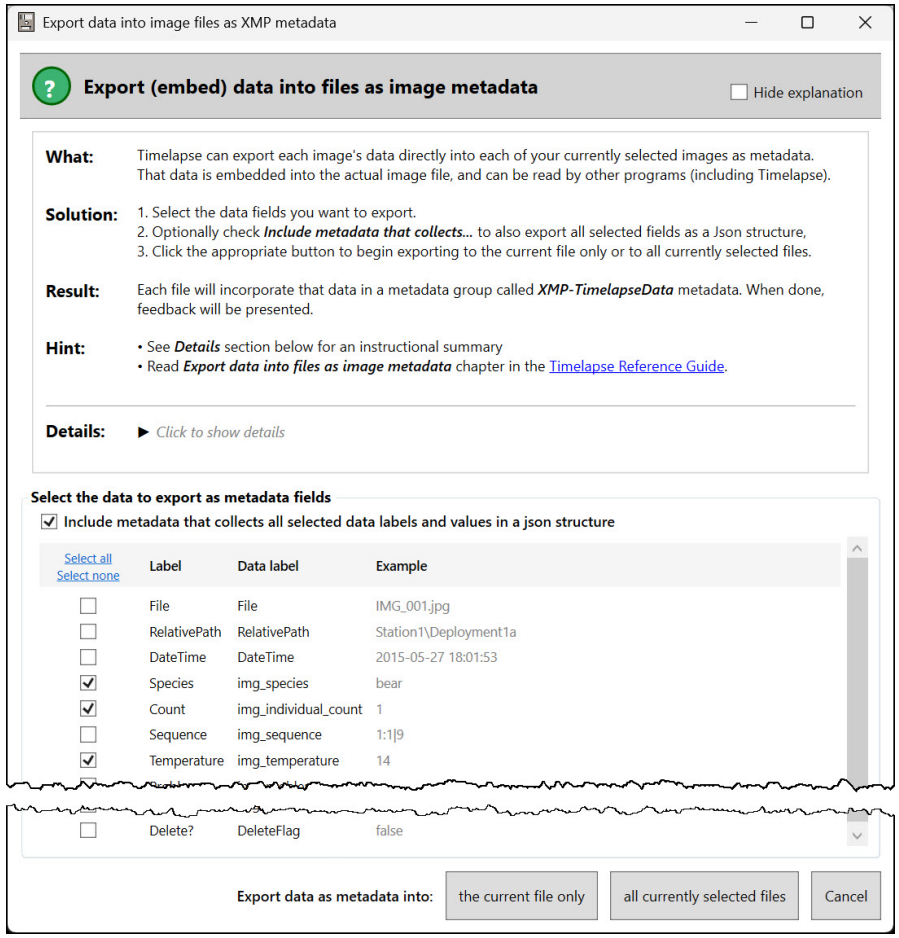
For example, if you export a data field whose data label is 'species' and whose value is 'deer', it will write this as a directory/Tag ID/value triad:

XMP-TimelapseData species deer

Important limitation. Timelapse cannot write metadata to **.AVI video** files. When exporting, **AVI** files will be skipped.

Timelapse Options for Writing Metadata

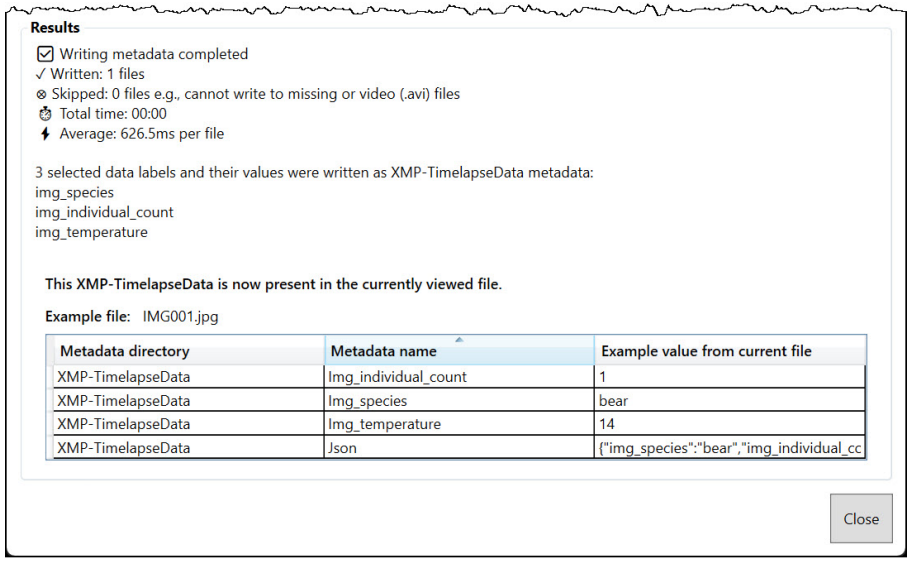
1. Open Timelapse on the practice image set, and display a file.
2. Select: ***File | Export (embed) data into files as image metadata***
   This dialog appears, where it lists the data label and label for each control. Example values are displayed, all taken from the current image.



3. Use the checkboxes to indicate which data fields you want to export. This example uses *Species*, *Count*, and *Temperature*.
4. If the ***Include metadata that collects all selected data labels...*** is checked, an extra metadata field will be written that collects all selected data labels in a json structure.

Writing image data as XMP-TimelapseData

5. Click ***the current file only*** to export metadata into the current file. (later, you can try exporting to all your currently selected files). A feedback panel similar to what is shown below will be displayed when exporting is completed.



As mentioned and as can be seen in the feedback panel, Timelapse writes custom metadata fields corresponding to the data fields whose checkbox was checked. We see:

- » directory = XMP-TimelapseData
- » tag-ID = the data label of the selected field (e.g., img_species)
- » value = the current value of that image's data field (e.g., bear).

We also see that a Json tag was created, which contains all the selected data label/value pairs in a json data structure.

Note. The above provides a one-to-one correspondence between the data field written and individual metadata entries. The tag is the same as the data label, and the value is the same as the data field's contents. This makes later data lookup and management much easier.

The json field provides an alternate way to read in metadata at a later date, as json is fairly easy to parse.

Video is more complicted

To make things somewhat confusing, writing data into video files can be somewhat problematic. In particular:

- *.avi* files
  - » Timelapse skips these files due to avi metadata-writing limitations,
- *.mp4* and *.mov* files
  - » ***IPTC:Keyword*** metadata cannot be written, as IPTC is meant for images, not videos
  - » ***XMP*** metadata is fine, including ***XMP-TimelapseData*** and the pre-defined ***XMP-dc:Subject*** metadata.
  - » This is why Timelapse tries to write pre-defined data into both the IPTC:Keyword and XMP-dc:Subject field.

Using that metadata

Once written, there are various means to actually use that metadata. However, you need to know that many off-the-shelf applications do not display XMP metadata.

As you will see below, Timelapse and other coding libraries often use ***Exiftool*** to view and extract XMP metadata.

Reading metadata using Timelapse

The Data Entry section have already described how you can populate Timelapse fields from a file's metadata via the ***Edit | Populate one or more fields with metadata***... menu item.

However, ***Metadata extractor*** (the default metadata viewer) does not show XMP metadata. To see it:

1. Click ***ExifTool checkbox*** to show XMP-TimelapseData metadata fields.
2. (optional) Click ***Details*** checkbox to show the Metadata directory.
3. (optional) Click ***Metadata directory*** column header to sort by Directory.

Reading metadata using R

R has an exiftool library that makes it fairly easy to read and extract XMP data. The R script below illustrates the few lines of code needed to read the XMP-TimelapseData as exported to our example image.

I ran this in R 4.5.2.

```
# Preparation: Install the exiftool and load the library
install.packages("exiftoolr")
library(exiftoolr)
Install_exiftool()

# Define a path to the image file using a 'raw string' to avoid \U errors
img_path <- r"(<your path goes here>/IMG_001.jpg)"

#  Get the metadata from the metadata group XMP-TimelapseData
#  We use the wildcard "*" to get all tags in that group
metadata <- exif_read(img_path, tags = "XMP-TimelapseData:*")

# Show the results
# Manipulating this list to get key value pairs is now just standard R stuff.
str(metadata_list)

# This is what would be printed with or example image
$ SourceFile : chr ".../IMG_001.jpg"
$ Img_individual_count: int 1
$ Img_species : chr "bear"
$ Img_temperature : int 14
$ Json  : chr "{\"img_species\":\"bear\",\"img_individual_count\":1,
          \"img_temperature\":\"14\"}"

#  We can also retrieve a specific tag by supplying it instead of the *, e.g.
json  <- exif_read(img_path, tags = "XMP-TimelapseData:json")
```

Reading metadata using Python

I haven't tried it, but just do an AI search on *How to read XMP metadata with Python*. It appears that many libraries are available.