Idempotent Photographical Categorization

Alex Homer 9 Mar 2014 12:37 AM | **0**

It's been many years since I switched from film to digital by selling my old Pentax SLR, extensive selection of quality lenses, and bag full of assorted attachments at some ludicrously low price. Since then my photographic arsenal has included several Olympus digicams. Yet I still haven't got the knack of successfully categorizing our ever-growing collection of photos.

At first it's easy, you just drop them into suitably named folders. Like most people, I suspect, I never quite get round to adding all the tags and other info that helps you search for photos. The problem comes as the collection grows. In our house, we use Media Center as the main TV, with a modified version of an old Coding4Fun screen saver sample (see "**The Screensaver Ate My Memory**") so that we get slideshows of photos at random when the system is idle. Yes, we actually get to see our photos regularly rather than them gathering virtual dust hidden away on a hard disk somewhere.

The screensaver presents them like the old Polaroid instant photos, with a caption containing the folder name and the date the photo was taken. However, increasingly I noticed that sometimes the date is wrong - usually because I fine-tuned the photo, scanned it from an old hard-copy print, or some friends sent it to us long after it was taken. But what really screwed things up was when, a few weeks ago, I was forced to reduce the total storage volume. I did it by running a macro in Paint Shop Pro that removes digital camera noise and shrinks most files by up to 60%.

As you can imagine, the result is that all of the photos now displayed that date, because - as I discovered by digging out the old source code - the screensaver reads the last-modified date of the file. No problem, I thought, just change it to read the created date instead.

If you haven't tried this, here's a tip: don't bother. I started off using the .NET File.GetCreationTime method, but that just gave some random result. So I dug back into the past and tried the old FileSystemObject we used to use in ASP scripts before the days of ASP.NET. And got the same result; obviously they use the same O/S functions. And if you get round to reading the blurb on the MSDN reference pages for the methods, you'll discover that you can't expect them to work. It says that NTFS caches the creation date, so it is only correct if you actually set it in code first - which is great if what you actually want to do is find out the date because you don't know what it is. Supposedly it only caches the value "for a short time", but waiting a day and rebooting the computer had no effect.

So, no problem, the cameras all know the date and time that the photo was taken and it will be in the EXIF properties of the file. Well it seems that everything you never really wanted to know, such as the aperture setting, shutter speed, quality setting, flash mode, lens manufacturer's name, and many other undecipherable values are there, but not the date and time - the field in the Origin settings for Date Taken is empty in every one. Err, why?

Ah, but the file name is a weird combination of letters and numbers (such as P0146752.jpg), which surely must be the date in some form of encoding. Well, after several hours looking at files, taking test photos, and playing Bletchley Park code-breaker, I couldn't figure it out.

In the end, I admitted defeat and decided that the obvious answer was to include some kind of tag in the filename that showed the month and year, and which could be easily extracted in the screensaver code for use in the caption. For some unaccountable reason I chose to add a tag of the form [t-MMM yyyy], so that the photos would have a filename such as P0356381[t-May 2013].jpg. It was easy to modify the screensaver to use the current folder name and the tag so I get a caption such as "Garden Birds May 2013"). The biggest job, of course, was going through all the photos adding the appropriate tag.

But it was worth it, now we get an accurate date for each photo and my wife tells me when I got one wrong. The nice thing is that whatever I do with the file in terms if modifying it, copying it to some device that doesn't properly handle dates, or some other so-far-unforeseen action, I will always have the correct date.

So, did marital harmony return to our house? Not quite. As my wife pointed out, when you try to view the photos in Media Center (or on any other connected device) they come out in some random order. The default alphanumeric filenames aren't in ascending order by date. So when you add new files to a folder, you have to search all through to find them. Oh dear.

What I should have done, of course, is put the date in the form yyyyMM at the start of the filename. But no problem, I can write a simple utility to rename the files automatically. In fact, I can even get it to both add both a suitable prefix (such as "201409") by reading the tag in the filename, as well as including an option to automatically generate the suffix tag for the screensaver by using the last-modified date of the file when I run it over new photos as I add them to our collection.

And, purely by luck, I've just finished working on our **Cloud Design Patterns** guide, which regularly reinforces the need to consider idempotency for operations that may be repeated. In my simple file renaming scenario, the issue is if I run the utility again over files that have already have a tag, or both a tag and a prefix. Obviously I don't want more tags and prefixes adding to the filename, so it's vital that the code checks if the filename already contains a tag before it creates one from the last-modified date, and only translates this into a prefix if the filename doesn't already contain one (I haven't got round to implementing any actions that would update a tag or prefix).

But after all this effort, I suppose I should have thought out the solution more thoroughly first, and just used the date prefix yyyyMM - and modified the screensaver to use that. But after a few days it occurred to me that I can put anything I like in the tag, not just a date, while the prefix will ensure that the photos still appear in ascending date order. So the effort wasn't totally wasted.

Though, afterwards, someone mentioned that I could just as easily have changed the name of the file to something meaningful and displayed that in the caption…