OPENMETA

January 16, 2009

Document management for OS X

a open sourced, simple, scalable metadata system that works

By Tom Andersen of ironic software

Document management is finding and storing your documents. Traditionally document management is done with intrusive, large 'systems'. The prices of these systems reflects their complexity.

For the small or not so small Mac based office, the current set of solutions is for the most part overkill.

We at Ironic Software have developed a system that is extensible, open sourced, and based on the OS X file system. Documents are not 'stored' in any special database, instead they are simply put where ever they make sense. A document management system must both store documents (which OpenMeta leaves up to the file system), and it must also store metadata about the files - meta data being tags, dates, ratings, people, etc associated with the document. OpenMeta uses the unix 'extended attributes' (xattr) to store this metadata.

By storing both the files and the metadata on the file system, backing up and restoring a document management system can be done with (almost) any normal backup system. For some offices, Time Machine will do more than an adequate job.

Storing metadata and files using no special formats or databases allows for very robust future proof behavior. OpenMeta has been designed to last as long as OS X will last. There are no proprietary databases to deal with.

With all the data stored on the file system, there does need to be a search mechanism - the metadata as well as the contents of the documents need to be able to be searched. Apple already has a powerful search engine running on OS X - Spotlight. By running straight forward queries in Spotlight, metadata may be searched on a field by field basis. Spotlight also of course allows searching across all metadata at once. The default search application that ships with OS X - namely Spotlight.app will often not be an adequate tool for users to construct searches and deal with documents. Products like Ironic's Deep are designed to offer a richer interface to document management than the simple Spotlight search UI.

January 16, 2009

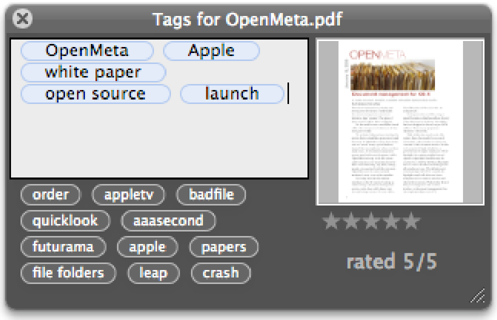
details

Why Open Source

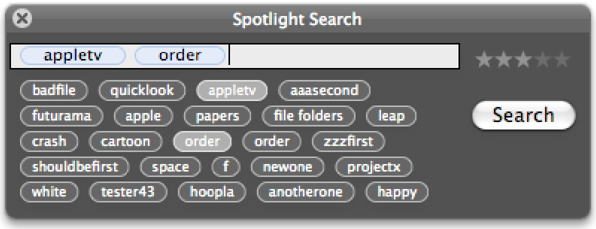
Open sourcing the ‘way’ to set metadata on a file in OS X allows for everyone to do it the same way, but just as important, sends a message to customers looking at an OpenMeta based system. That message being of course that us vendors will have much less ‘lock-in’ than with a wholly proprietary system. What is open source in OpenMeta *is* the formats, attribute names and conventions for metadata storage. Along with source code to store, retrieve and in some cases validate the information stored. OpenMeta in and of itself will not be a solution for anyone but the largest companies willing to do in house development of the tools and systems appropriate for their business.

Tools for the job

Ironic is releasing a simple command line tool omtool, along with a simple OS X application called ‘Tagger’ (shown here) which allows users to tag and rate any file(s) on their system. Tagger also generates searches for files with ratings and tags as supplied in it’s search box.



In addition to these free utilities, Ironic has also shipped Deep - an image manager, and will soon be transitioning both Yep and Leap to OpenMeta. Deep uses OpenMeta to store tags and also to store color information (palettes) about each image on a computer. In writing Deep we have developed the OpenMeta code to be able to handle the demands of a real commercial application.



Technical details.

The OpenMeta project is hosted at http://code.google.com/p/openmeta , at that site there is source code, a forum, a wiki, and more. A few points:

- OpenMeta uses no ‘secret’ Apple API.
- Indexing of Spotlight-able data is automatically handled by the OS.
- It is easy to add new metadata keys - these keys can be kept private for internal use, or registered in OpenMeta in order to build consensus. example: Workflows.
- Complex, solution specific binary data can also be stored.
- OpenMeta uses setxattr()/getxattr() to get and set all metadata.
- Because the metadata is not ‘inside the file’ it is easy to develop tools that allow metadata to be included / excluded in a export. For example, when emailing a document, it may be desirable to not send along the private metadata that is set on the file.

- Access rights by users to files is fundamentally determined by file permissions on the server/desktop/laptop that has the files resident on it. There are no other systems. If a user can log into a file system - then they can get at the files on it. Other document management solutions have their own built in access permissions system. While this can be useful, there are more times than not when this adds an unneeded layer of complexity.

The competition

Is this not already done? OpenMeta compared to other tagging solutions

Doesn't Spotlight already index tags?

Spotlight's job is to index all the files on a computer and put the relevant data that it finds into a database, so that searching can be done. Spotlight does not store any data in its database. In fact you can (and should monthly) wipe out the Spotlight database on a computer - OS X will simply rebuild it over a manner of hours. For instance, Spotlight does find keywords and ratings on some files, and dutifully stores them in the database for searching against. The big problem with Spotlight as it stands is that there is no way for a user to add their own meta data to an arbitrary file.

Enter XMP, spotlight comments, and others

It turns out that some file formats have various sections inside them where metadata can be stored. IPTC/Exif metadata on some image formats is a good example of this. When Spotlight looks at an image while building its database, it will parse the IPTC/Exif data on the image, incorporating that data into the Spotlight DB for searching. This includes such important data such as exposure time, focal length, etc. Standards have risen for the storage of tags (aka keywords) in the IPTC data. This is a good thing. Adobe has taken this metaphor of storing metadata inside a file to the max with Adobe Bridge and related products. With Bridge, you can add tags and other meta data to virtually any document, which sounds good. The 'big but' in all of this is of course not every file format supports embedded metadata, and even for those that do, the process of reading and writing this metadata involves editing the actual file in question, which can sometimes lead to problems. For files that don't have any metadata storage facility, Adobe has to store the data somewhere - it usually ends up in so called 'sidecar' files.

Problems with an XMP - like approach:

- Metadata is stored in multiple locations - different places inside different file types, using different wrapper formats, etc. Often the data is not stored in the file, but in some 'sidecar' file.
- Keywords and other metadata that 'shipped' with a document are then usually mixed in with metadata set in the office. This can cause problems, as some files have literally hundreds of keywords attached to them - this can water down the effectiveness of a document management system.
- When sending files all metadata set on files is often included, with little prospect for removing it, unless XMP aware applications are used to edit documents before sending.

- The metadata format that XMP is written to is very complex, and not easily understood by IT personnel (like for instance me). The source code to implement XMP shows just how complex the entire process is.
- Having said all of that - it is obvious that recording the exposure time right into a photo is a ‘good thing’. OpenMeta allows users to search through the contents of a file - and through user set metadata.

Tagging, Spotlight comments, and a short history

Ironic Software has been writing document management software for three years. Yep, our first application, allows users to add tags to pdf documents. We looked at the three methods that we could think of to set tags on PDF documents: Each of these ‘looks’ used up untold hours of writing, testing and debugging.

1) PDF documents have a provision for tags inside them. Apple has an API for setting these PDF keywords. Sounds good. Problems: There were bugs (now largely solved?) in the Apple software - setting tags would often eliminate the table of contents in a PDF, and sometimes the PDF could get corrupted. Also setting the tags on hundreds of documents at once was very slow, as each file had to be read in and written out.

2) Use ‘Spotlight comments’ to store the tags. It turned out that there are several tagging solutions based on this for the Mac. Problem: There are no standards for how tags are to be stored in the comments, especially if the tags have to share the comment with other ‘actual comments’. Another problem is that Spotlight comments are stored in .DS_Store files, which are an ancient holdover from system 8. This in itself caused enormous problems, as the Finder struggled to keep up with changes. (If I could have a nickel for each time I cursed those Spotlight comment based tags!)

3) Use our own database to store tags. This is what we settled on, knowing or rather hoping that we could figure out something more robust in the future.

It is our hope that OpenMeta will allow organizations from the size of a single MacBook all the way up to large corporations to organize, search and collaborate documents using simple user centered tools.

The OpenMeta project is hosted at http://code.google.com/p/openmeta , at that site there is source code, a forum, a wiki, and more.

Ironic software’s web site is at http://www.ironicsoftware.com , and Tom Andersen can be reached at tom.andersen@gmail.com