Embedding metadata in PDF finding aids to enhance discoverability

Author: Elizabeth Post

August 2016

Embedding Metadata in PDF Finding Aids to Enhance Discoverability

Betsy Post, Boston College Libraries

August 2016

Project Overview

The Boston College Libraries recently completed a project to enhance discoverability of our finding aids. We migrated the finding aids from a repository to a web server and created a sitemap for ArchiveGrid and Google harvesting.

All of our finding aids are in PDF format, a format we prefer for a number of reasons. Our users are accustomed to using PDF finding aids and like the way it prints. Choosing PDF allows us to deliver both legacy finding aids created using word processing programs and newer finding aids created in the Archivists' Toolkit in a consistent format.

Because we have rich metadata containing key identifiers in all three of the systems that we needed to execute the project (ALMA, Archivists' Toolkit, and Digitool), we thought our migration plan would be quick and without complications. With minimal effort expended, we selected a server, added some finding aids, generated a test sitemap, and asked Bruce Washburn at ArchiveGrid to do a test harvest. That's when things started to get a little more complicated.

Our sitemap worked well, but Bruce's helpful feedback included a zinger. The "zinger" was a direct consequence of our preference for the PDF format. To optimize our finding aids for ArchiveGrid and search engines such as Google, we needed to embed metadata about each finding aid into the document's header. Here is how Bruce explained the problem:

> There is one issue though, when the sitemap is used in conjunction with PDF finding aids. For other document formats (EAD XML or HTML) I can usually find a distinctive title and sometimes a scope and content note, just by examining the document markup. PDF documents have a capacity for including document properties for its title, author, subject, and other customized elements, but these need to be added as part of the PDF document creation … unlike EAD XML and HTML, they aren't an automatic by-product of the document's internal structure.

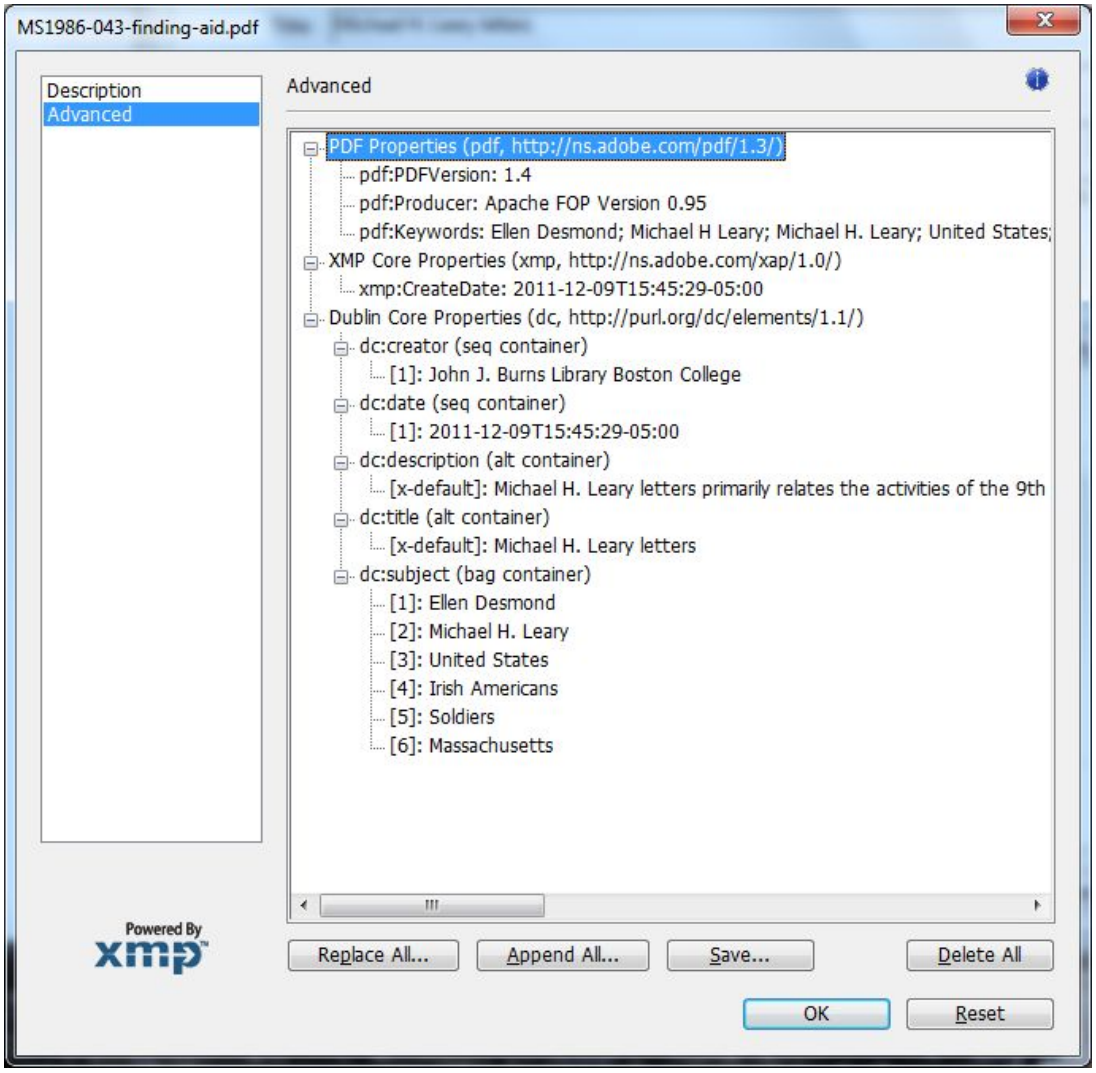
Diving Into PDF Properties

While we had all noticed the presence of the PDF metadata properties that can be viewed in Acrobat by clicking on File->Properties, we faced two challenges:

1. We are fortunate enough to have Adobe Acrobat Pro and can edit the metadata properties as a manual process, but wanted more efficient solutions for both our retrospective and current workflows.

2. Although we have experience with Dublin Core, we didn't have much experience with either the Adobe PDF Properties or XMP schemas. (See Figure 1)

Figure 1: PDF Properties (Additional Metadata/Advanced)



We hypothesized that we could avoid manual editing for new finding aids by adding a few lines of code to the xsl:fo that the Archivists' Toolkit uses to generate PDF finding aids. For the retrospective collection, we could use Image::ExifTool, a PERL module, to write metadata extracted from MARC records into the PDF properties.

While technical answers jumped to mind pretty quickly, we still had our second problem. We didn't know much about Adobe PDF Properties or XMP schemas. We began tentatively by using the File->Properties templates and observing what happened when we edited the fields in the primary "Description" and "Additional Metadata" forms (see figures 2 and 3). We also reviewed the information in the document PDF/A Metadata XMP, RDF & Dublin Core.

Figure 2: PDF Document Properties Description Form

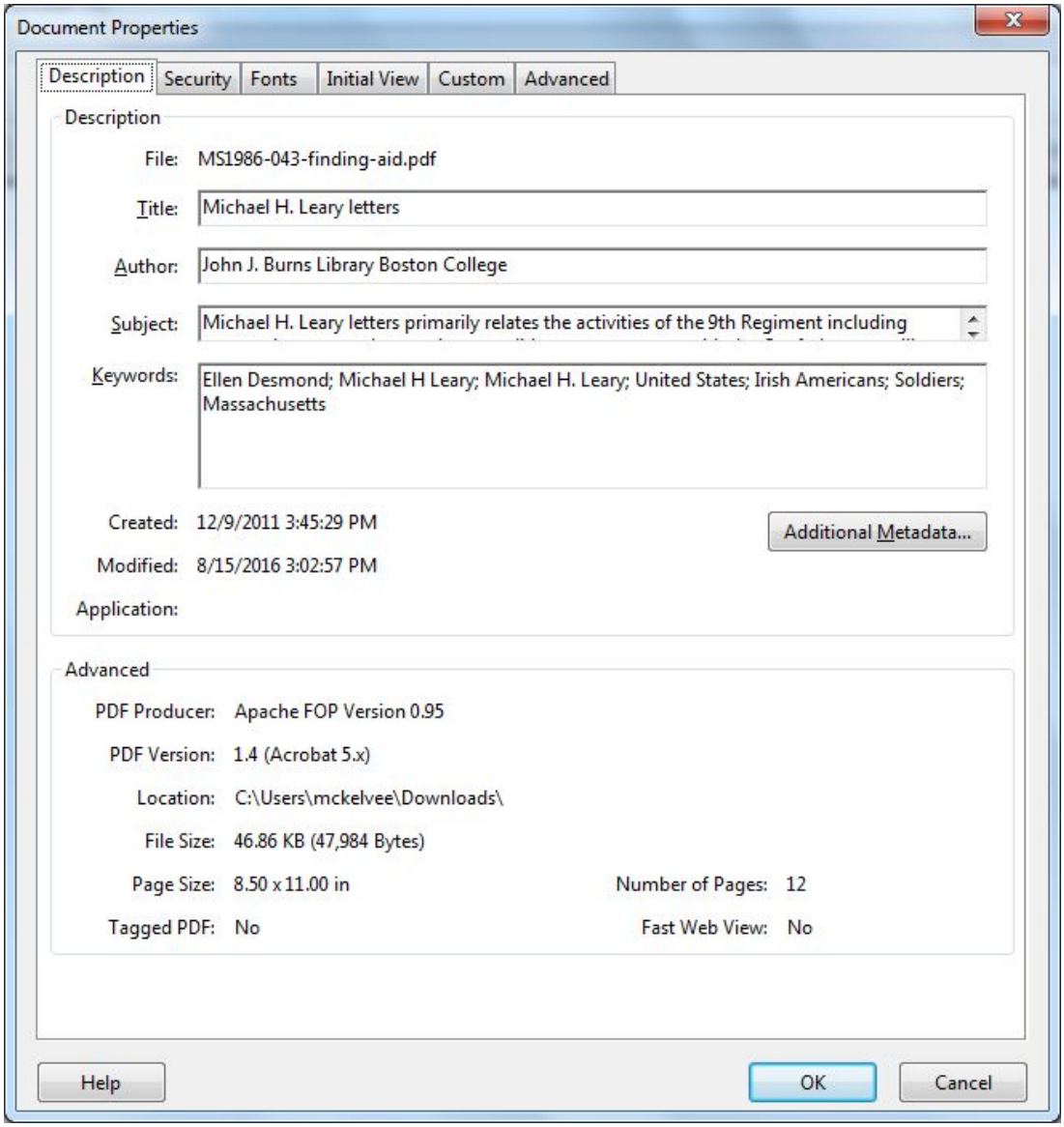
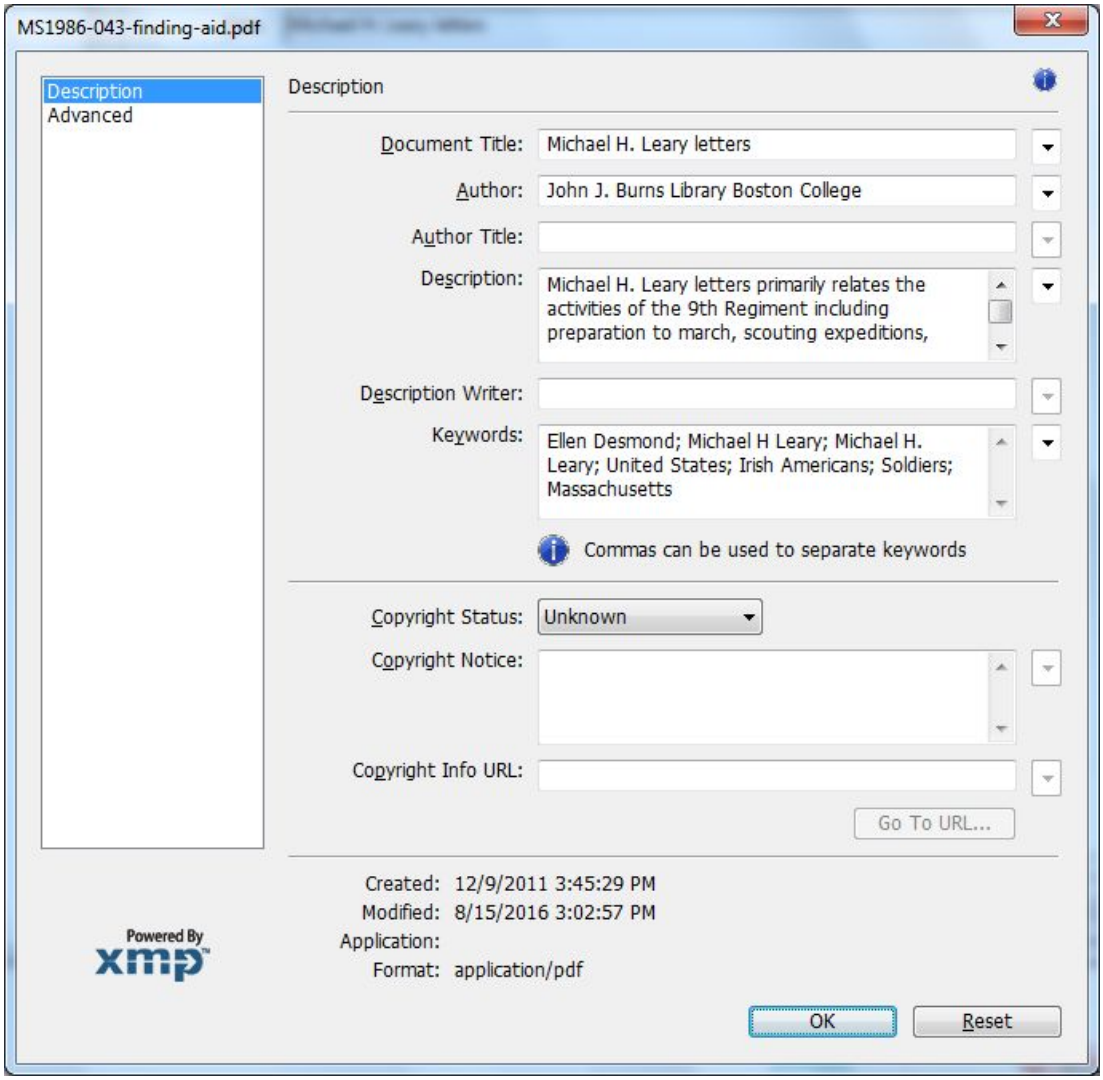


Figure 3: PDF Document Properties Description Form - Additional Metadata



Our experimentation resulted in a couple of surprises about the behavior of the subject and keywords fields in the template.

PDF Property Surprises

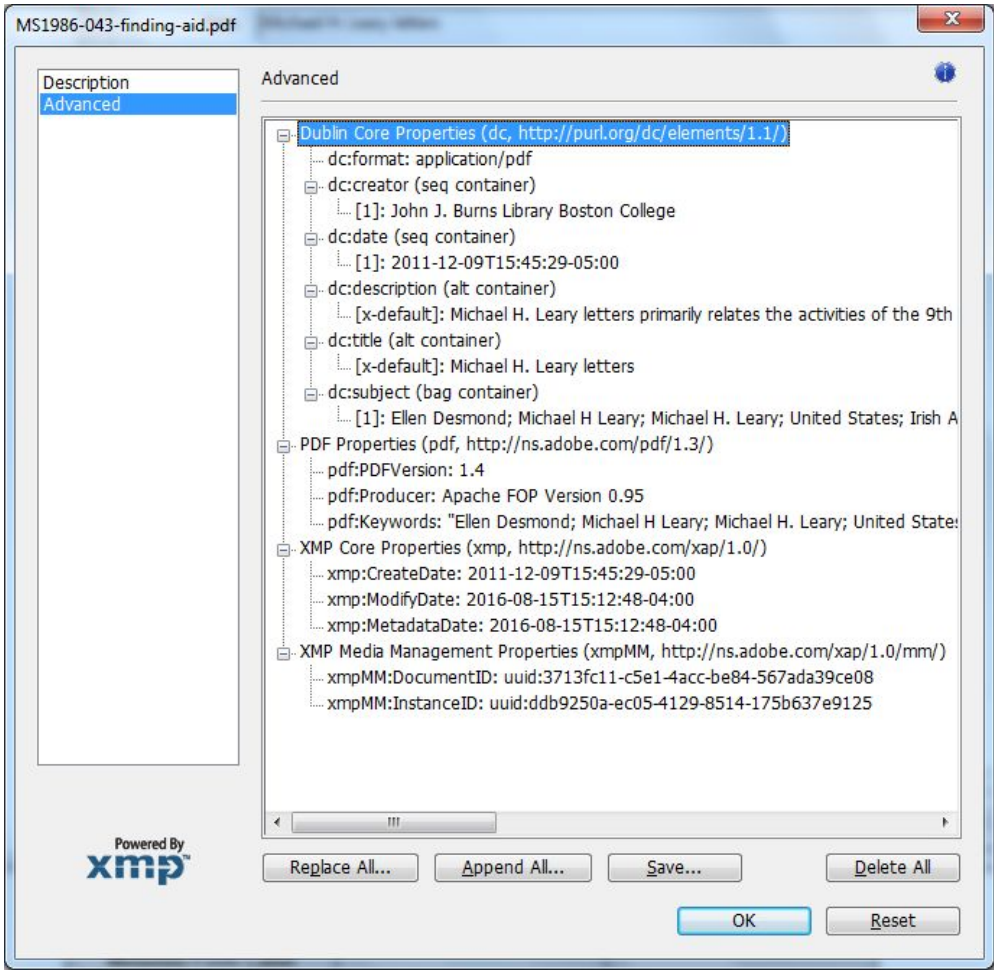
Our first surprise involved the *Subject* field in the template (see Figure 2). Our first guess was that the subject field should be used for subject headings of some sort or another. Further testing suggested that this is not the case. We learned that the *Subject* field actual corresponds to dc:description and thus felt it was the most appropriate place for the abstract (compare figures 2 and 1).

The second, and slightly more complicated surprise, had to do with keywords. It seemed odd that both the *Description* and the *Additional Metadata* templates had a field for keywords (see figures 2 and 3). Initially, the fields seemed identical. Updating metadata on one form changed it on the other. But why would Adobe put a second, redundant keywords field on its form?

A little more experimentation showed that the fields were different and that the differences had implications for our workflows. Even though they contain duplicate text, one of the keywords fields maps to pdf:keywords and the other maps to dc:subject. Figure 1 shows the pdf:keywords and the dc:subject fields containing the same terms. However, in the PDF keywords field, all of the keywords are in a single semicolon delimited string. The dc:subject is repeated, with one keyword per element.

The tricky relationship between dc:subject and pdf:keywords had design implications for our script that adds metadata to our PDF files. Writing to one of these elements overwrites the data in the other. We found that if we wrote to pdf:keywords then our dc:subjects were incorrectly formatted into a single semicolon delimited string as shown in Figure 4 below. To fix this problem, our script writes each keyword phrase into a separate dc:subject and lets the behind-the-scenes Adobe magic generate the concatenated pdf:keywords.

Figure 4: Badly Formatted dc:subjects element



Results: PDF Properties Crosswalk, File Naming Conventions, and Scripts!

After some study, we created a crosswalk to inform how our scripts would write metadata to our PDF files.

Table 1: Mapping Metadata to PDF Properties

PDF Document Properties Metadata Form Label	Retrospective	Prospective
Title	MARC 245|a	Use the same stylesheet logic that Archivists' Toolkit uses to add the title to the finding aid cover page
Author	Boston College John J. Burns Library	Boston College John J. Burns Library
Subject	MARC 520|a	/ead:ead/ead:archdesc/ead:did/ead:abstract
Keywords	MARC 6xx|a ● Invert personal names, eliminating the comma ● Strip periods from the end of 6xx|a ● De-dupe identical 6xx|a fields ● Output as a semicolon delimited list, stripping leading and trailing spaces and tabs	While it is possible to pull subject headings from the EAD, we don't add MARC subject headings to our finding aids. For now, we are copying and pasting this information from the MARC record. This manual process follows the same logic as documented for our retrospective script in the previous column. To preserve our work, we add an internal only note in the Archivists' Toolkit resource record.

When our finding aids were in a repository, file naming didn't seem critical as the files received ID numbers in the repository; however, on a web server, systematic file naming that provides enough information to identify the finding aid without opening it is critical. Our finding aids naming conventions appear in Table 2. A script was used to retrospectively rename the finding aids in the repository.

Table 2: FileNaming Conventions

FileName = “Resource id (special characters stripped or replaced)” + “-” + “finding-aid.pdf” Example: MS2000-008-finding-aid.pdf

---

Adding Metadata to Existing PDFs

As noted earlier, we had good metadata about our finding aids. To create a tab delimited file of the metadata we wanted to embed in the PDF finding aids, we ran a script *metadata.pl* on a file of MARC records that had related finding aids. We then enhanced the tab delimited file by adding the current finding aid file name to the beginning of each row (using a temporary database). Finally we ran *addMDtoPDF.pl* to embed the metadata from the tab delimited file into the PDF finding aids. Our scripts are available on GitHub.

Adding Metadata to New PDFs

We modified the the Archivists’ Toolkit *at_eadToPDF.xsl* stylesheet so that metadata would be embedded in our PDFs at the time of export. Our changes to the PDF stylesheet allow for title, abstract, and author to be embedded in the PDF metadata properties. Because of time and workflow considerations, we do not include MARC subject headings in our EAD. Thus, for new finding aids, MARC headings are manually copied from our catalog into the PDF properties. Our version of *at_eadToPDF.xsl*, which belongs in the Archivists’ Toolkit folder Program Files\Archivists Toolkit 2.0\reports\Resources\eadToPdf is available on GitHub. PDF properties are handled in lines 199-258.

Icing On The Cake

Once all our finding aids were on the new server and good to go, our systems department created two important chron jobs that make the process very smooth. One of the chron jobs updates the sitemap every few minutes. A second chron job updates the permissions on the files every few minutes. This safeguard ensures that one staff member doesn’t accidentally upload a PDF with file permissions that would prevent another staff member from overwriting the file with an updated version.

Members of the Boston College Finding Aid Discoverability Team
Amy Braitsch, Brian Meuse, Betsy Post, Emily Singley, Beth Sweeney, Kelly Webster