AcroTeX.Net

The aebXMP Package

Updating XMP using E4X and LaTeX

D. P. Story

Package Components
Click on the brown links to save to your hard drive.

- Save aebxmp.sty, the package file
- Save aebxmp_test.tex, a simple test file

Last Revision Date: March 12, 2007

Table of Contents

1. Introduction

The motivation for the development of this package came from Herr Jürgen Gilg, who had a need to fill in the metadata fields beyond those normally populated by using hyperref: Title, Author, Subject and Keywords. Of particular interest to him were the metadata fields Copyright Status, Copyright Notice and Copyright Info URL.

After doing some research on the CTAN archives, I came across the hyperxmp package by Scott Pakin.[1] The package works well with pdftex and dvipdfm, but has a bit of a problem when using the distiller. For this reason, I sought my own solution to the problem.

As a beta tester of Acrobat 8 Professional, I had the opportunity to use some of the new capabilities of the JavaScript interpreter as an alternate approach to the one used by Mr. Patkin. The JavaScript version 1.6 interpreter, the one used in version 8, comes with E4X, an XML parser, built in. I could see that E4X could be exploited to manipulate the XMP data, and this was my approach.

2. Requirements

The techniques used by the aebxmp package to update the XMP data require the AcroTEX eDucation Bundle (AeB), freely available from `www.acrotex.net`. Because E4X is used, we also require Acrobat 8 Professional, and, since you have Acrobat 8, my LaTeXing friend, use Acrobat Distiller to create your PDF.

3. Acquiring aebxmp.sty and its test file

The package aebxmp.sty and a simple test file aebxmp_test.tex are included as attachments to this demo file. Click the links on the cover page of this document to save the components of this package.

To use this package, you must have, in addition to Acrobat 8 Pro, installed on your computer a standard TeX system, including the latest version of AeB.[2]

4. New LaTeX commands

As mentioned previously, the aebxmp package addresses three areas of interest: Setting the Copyright Status, Copyright Notice, and the Copyright Info URL. Obviously, other elements of the XMP can be addressed. To that end, the aebxmp package defines three new LaTeX commands to populate the values of the three metadata fields Copyright Status, Copyright Notice and Copyright Info URL.

[1] The reader is invited to read the documentation of the hyperxmp, as contained therein is a good discussion of XMP (eXtensible Metadata Platform).

[2] AeB can be downloaded form `http://www.acrotex.net`.

```
\newcommand{\copyrightStatus}[1]{\def\webcopyrightStatus{#1}}
\let\webcopyrightStatus\@empty
\newcommand{\copyrightNotice}[1]{\def\webcopyrightNotice{#1}}
\let\webcopyrightNotice\@empty
\newcommand{\copyrightInfoURL}[1]{\def\webcopyrightInfoURL{#1}}
\let\webcopyrightInfoURL\@empty
```

Values for the arguments of these commands are documented below.

`\copyrightStatus{True|False}`: If `True` the Copyright Status is set to `Copyrighted`; if `False`, Copyright Status is set to Public Domain. If left empty, the status is set to Unknown.

`\copyrightNotice{<text>}`: The `<text>` of the Copyright Notice is defined

`\copyrightInfoURL{<URL>}`: The `<URL>` to the copyright information

For example:

```
\copyrightStatus{True}
\copyrightNotice{Copyright (C) 2006-\the\year, D. P. Story}
\copyrightInfoURL{http://www.acrotex.net}
```

Enter unicode (`\uXXXX`) directly into the `\copyrightNotice` and `\copyrightInfoURL` fields; for example,

```
\copyrightNotice{Copyright (C) 2006-\the\year, J\u00FCrgen Gilg}
```

5. Updating the XMP

The basic methodology is as follows:

- Acquire the metadata using the JavaScript property `this.metadata`, and convert the returned string to XML via the `XML` constructor.
- Use E4X to add the appropriate elements to the XMP Data.
- Save back to the document by converting the XML to a string using the `toXMLString()` method, then assigning this string to `this.metadata`.

The JavaScript for updating the XMP data is introduced using the execJS environment of the AeB. The verbatim listing of the script is included for your reading pleasure:

```
\begin{execJS}{execXMP}
    var meta = this.metadata;
    var myXMPData = new XML(meta);
    var myx = new Namespace("adobe:ns:meta/");
    var myrdf = new Namespace("http://www.w3.org/1999/02/22-rdf-syntax-ns#");
    var mypdf = new Namespace("http://ns.adobe.com/pdf/1.3/");
    var myxap = new Namespace("http://ns.adobe.com/xap/1.0/");
```

```
    var mydc = new Namespace("http://purl.org/dc/elements/1.1/");
    var myxapRights = new Namespace("http://ns.adobe.com/xap/1.0/rights/");
    var p = myXMPData.myrdf::RDF.myrdf::Description;
/*
    We test whether this element has a value already, if no, we assign it a value,
    otherwise we assign it another value.
*/
    if (p.mydc::rights.myrdf::Alt.myrdf::li.toString() == "") {
        p[0] +=  <rdf:Description rdf:about=""
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
            <dc:rights>
            <rdf:Alt>
                <rdf:li xml:lang="x-default">{"\webcopyrightNotice"}</rdf:li>
            </rdf:Alt>
            </dc:rights>
        </rdf:Description>
    } else
        p.mydc::rights.myrdf::Alt.myrdf::li = "\webcopyrightNotice";
/*
    We save xapRights:Marked and xap:WebStatement, delete old values,
    then replace them with the new values.
*/
    var saveMarked = p.@myxapRights::Marked.toString();
    var saveWebStatement = p.@myxapRights::WebStatement.toString();
    delete p.@myxapRights::Marked;
    delete p.@myxapRights::WebStatement;

    p[0] +=  <rdf:Description rdf:about=""
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:xapRights="http://ns.adobe.com/xap/1.0/rights/">
            <xapRights:Marked>\webcopyrightStatus</xapRights:Marked>
            <xapRights:WebStatement>{"\webcopyrightInfoURL"}</xapRights:WebStatement>
    </rdf:Description>

//  Convert myXMPData into a string
    myNewXMPStr=myXMPData.toXMLString();
// and assign it to the document metadata
    this.metadata = myNewXMPStr;
\end{execJS}
```

6. Checking for validity

While looking at this file in Acrobat (or Adobe Reader), press `Ctrl+D` to get the Document Properties dialog box. Select the Description tab and click Additional Metadata. Since this document was built using the aebxmp package, with the declarations

```
\copyrightStatus{True}
```

```
\copyrightNotice{Copyright (C) 2006-\the\year, D. P. Story}
\copyrightInfoURL{http://www.acrotex.net}
```

in the preamble, you should see the following entries:

Copyright Status: `Copyrighted`

Copyright Notice: `Copyright (C) 2006-2007, D. P. Story`

Copyright Info URL: `http://www.acrotex.net`

I promise you that I did not enter these values through the user interface. `:-)`

7. Resources

The resources for the development of this package are

- *Standard ECMA-357: ECMAScript for XML (E4X) Specification*, http://www.ecma-international.org/publications/standards/Ecma-357.htm
- *XMP Specification*, http://www.adobe.com/go/acrobat_developer
- *Acrobat JavaScript Scripting Reference*, Version 8.0 http://www.adobe.com/go/acrobat_developer
- hyperxmp package by Scott Pakin. ftp://cam.ctan.org/tex-archive/macros/latex/contrib/hyperxmp/
- The AcroTEX System Tools, available for free download at www.acrotex.net. This is a LATEX-based system.

Now, I simply must get back to my retirement. 𝔇𝔓𝔖