

AcroTeX.Net

The aebXMP Package
Updating XMP using E4X and L^AT_EX

D. P. Story

Copyright © 2012 dpstory@acrotex.net
Prepared: January 10, 2012

Table of Contents

| | | |
|----------|--------------------------------------|----------|
| 1 | Introduction | 3 |
| 2 | Requirements | 3 |
| 3 | The Test File | 3 |
| 4 | Metadata | 3 |
| 4.1 | Author, Author | 4 |
| 4.2 | The \Keywords Command | 4 |
| 4.3 | Additional Metadata | 5 |
| 4.4 | Custom Document Properties | 6 |
| 5 | Checking for validity | 7 |
| 6 | Resources | 7 |

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 dvi_{ps}dfm, 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 L^AT_EXing friend, use Acrobat Distiller to create your PDF.

3. The Test File

The package aebxmp has a simple test file, aebxmp_test.tex, which is found in the examples folder.

To use this package, you must have, in addition to Acrobat 8 Pro (or later), installed on your computer a standard T_EX system, including the latest version of AeB.²

4. Metadata

The hyperref package provides basic metadata support, providing a mechanism for providing the title, author, subject, and keywords. Beyond these, we can include additional metadata.

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

²AeB can be downloaded from any CTAN server, from www.math.uakron.edu/~dpstory/webeq.html, or from www.acrotex.net.

4.1. Author, Author

The `hyperref` package allows you to specify an author or authors; the `Web` package uses this through the `\title` command, which passes its argument on to `hyperref` of inclusion in the PDF Info dictionary. The authors names are not individually assessable through JavaScript. In Acrobat 9, I believe, the JavaScript API includes the `Doc.info.Authors` property, its value is an array of authors, We can access each author using array notation: First author is `this.info.Authors[0]`, second author is `this.info.Authors[1]`, and so on. The number of authors is obtained using the `length` property of arrays, `this.info.Authors.length`.

The `AeBXML` interface to this is through the `\Authors` command. It takes a list of authors, each enclosed in braces, no commas (,) between the authors. Like so:

```
\Authors{D. P. Story}{J\u00FCrgen Gilg}
```

Using `\Authors` will overwrite the author(s) named in the `\title` command (of `Web`), or more generally, passed to `\hypersetup{pdfauthor={<author>}}` of `hyperref`.³

If you do use `\Authors` and overwrite the author(s) as passed through `hyperref`, the first author listed will be the one returned by the JavaScript command `this.info.Author`; in the Document Properties Dialog Box (Ctrl+D), all authors are listed in a semi-colon delimited list.

4.2. The \Keywords Command

The `\Authors` command utilizes `Doc.info.Authors`, which takes an array of authors names. There is no such property available for `Doc.info.Keywords`, the value of this property takes only a string of keywords. The keywords are stored by Acrobat in three ways, (1) `pdf:Keywords`; (2) in the **Info** dictionary; and (3) in `dc:subject`. In the latter case, `dc:subject` takes a Bag of subjects (keywords). A Bag, in XMP parlance, is an unordered array.

The command `\Keywords` takes a comma delimited list of keywords. Notice the word is capitalized to distinguish it from `\keywords`, which is defined in the `Web` package as the interface to inserting the keywords, via `hyperref`, into the **Info** dictionary.

```
\Keywords{AcroTeX.Net,XMP,E4X,Adobe Acrobat,JavaScript}
```

The command takes each comma-delimited list of keywords and inserts each into the `dc:subject` part of the metadata. For this document, the keywords appear in the XML metadata as

```
<dc:subject>
  <rdf:Bag>
    <rdf:li>AcroTeX.Net</rdf:li>
```

³This may be to your advantage when using `Web`; the value of `\title` is used to display the author(s) of the document, you may want to present the names one way on the title page, for example, and another way in the Description tab of the Document Properties dialog box.

```

<rdf:li>XMP</rdf:li>
<rdf:li>E4X</rdf:li>
<rdf:li>Adobe Acrobat</rdf:li>
<rdf:li>JavaScript</rdf:li>
</rdf:Bag>
</dc:subject>

```

When the document is first opened, the following command is executed

```
this.info.Keywords="AcroTeX.Net; XMP; E4X; Adobe Acrobat; JavaScript"
```

Here, I have broken the string across lines for readability. This inserts the same list of keywords into the **Info** dictionary.

When you executed this `this.info.Keywords` in the console, you'll get

```
AcroTeX.Net; XMP; E4X; Adobe Acrobat; JavaScript
```

as expected.

To access the individual keywords, I've inserted a document-level function, `aKeywords`. It takes as its argument the index of the keyword you want to get; For example, executing `aKeyword(0)` in the console returns a value of "AcroTeX.Net", while `aKeyword(4)` returns a value of "JavaScript". If you execute `aKeyword(5)`, a value of `undefined` is returned.

Listing the keywords may be of interest to someone, it is an exercise to me.⁴

4.3. Additional Metadata

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 five new \LaTeX commands to populate the values of the five metadata fields Copyright Status, Copyright Notice, Copyright Info URL, Author Title, and Writer Description.

Values for the arguments of these commands are documented below.

`\copyrightStatus{True|False}`: If `True`, 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

`\authortitle{<text>}`: The `<text>` appears in the Author Title line on the Additional Metadata dialog box. This is a **Photoshop** property. (See the Advanced category in the left panel.)

⁴Actually, this is the way **Acrobat** handles a comma-delimited list of keywords when the words are entered through the user interface; it puts them in a Bag.

`\descriptionwriter{<text>}`: The `<text>` appears in the Description Writer line on the Additional Metadata dialog box. This is a **Photoshop** property. (See the Advanced category in the left panel.)

For example, for this document, we have in the preamble,

```
\copyrightStatus{True}
\copyrightNotice{Copyright (C) 2006-\the\year, D. P. Story}
\copyrightInfoURL{http://www.acrotex.net}
\authortitle{Programming and Development, AcroTeX.Net}
\descriptionwriter{Testing and Promotions Department, AcroTeX.Net}
```

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

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

Unicode can be used in `\authortitle` and `\descriptionwriter` as well.

4.4. Custom Document Properties

You can define custom properties using the `\customProperties` command.

```
\customProperties
{%
  {name=<name1>,value=<value1>}
  {name=<name2>,value=<value2>}
}
```

The value of the name key requires a unique name, which must not be one of the standard property names Title, Author, Subject, Keywords, Creator, Producer, CreationDate, ModDate, and Trapped. The value of name shall consist of the characters A–Z, a–z, and 0–9, and beginning with a letter. The value may contain unicode, for example, in the preamble of this document we have,

```
\customProperties
{%
  {name=Developer,value={D. P. Story, Esq.}}
  {name=Motivator,value=J\u00FCrgen Gilg}
}
```

The properties may be accessed through the `info` property of the Doc object. The button (on the left) opens the console debugger window and displays all the document properties.

The custom properties may be viewed using the user interface; press `Ctrl+D` and choose the Custom tab.

For more information on this topic, see [Part 3, Storage in Files](#), section 3.2 on **Native metadata in PDF files**, in particular, see section 3.2.1 concerning user-defined document properties.

5. 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}
\authorTitle{Programming and Development, AcroTeX.Net}
\descriptionwriter{Testing and Promotions Department, AcroTeX.Net}
```

In the Advanced Metadata dialog box, you should see,

```
Copyright Status: Copyrighted
Copyright Notice: Copyright (C) 2006-2012, D. P. Story
Copyright Info URL: http://www.acrotex.net
Author Title: Programming and Development, AcroTeX.Net
Description Writer: Testing and Promotions Department, AcroTeX.Net
```

in addition to the usual Document Title, Author, Description, and Keywords. I promise you that I did not enter these values through the user interface. :-)

6. 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/devnet/xmp.html>
- *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 L^AT_EX-based system.

Now, I simply must get back to my retirement. ~~DS~~