Generation of PDF/X-1a and PDF/A-1b compliant PDF's with PDFTEX — pdfx-ext.sty

C. V. RADHAKRISHNAN and HÀN THÊ THÀNH

QUICK LINKS

1. Introduction

PDF/X and PDF/A are umbrella terms used to denote several ISO standards that define different subsets of the PDF standard. The objective of PDF/X is to facilitate graphics exchange between document creator and printer and therefore, has all requirements related to printing. For instance, in PDF/X-1a, all fonts need to be embedded and all images need to be CMYK or spot colors. PDF/X-2 and PDF/X-3 accept calibrated RGB and CIELAB colors along with all other specifications of PDF/X-1a.

PDF/A defines a profile for archiving PDF documents which ensures the documents can be reproduced the exact same way in years to come, a key element to achieve this is that the PDF/A documents shall be 100% self contained. All the information needed to display the document in the same manner every time is embedded in the file. A PDF/A document is not permitted to be reliant on information from external sources. Other restrictions include avoidance of audio/video content, JavaScript and encryption. Mandatory inclusion of fonts, color profile and standards based metadata are absolutely essential for PDF/A.

This package currently supports generation of PDF/X-1a and PDF/A-1b compliant documents using PDFTEX. More standards will be included in future.

2. Usage

The file, namely `pdfx-ext.dtx` is a composite document of program code and documentation in LATEX format in the tradition of *literate programming*. You can extract the program code alone by stripping off the documentation part by running LATEX or TEX over the installer namely, `pdfx-ext.ins` which is also provided with this file. To get the documentation which you are reading now, you need to run (PDF)LATEX over the file, `pdfx-ext.dtx`.

2.1. Data file for XMP metadata

As mentioned above, standards compliant PDF documents need XMP metadata to be included. In order to create XMP in the prescribed XML format, a simple data file holding the meta information of the document needs to be created either through a program or by hand. For our purposes, we name it as `\jobname.xmpdata`, a simple example of which will look like the following:

```
\Keywords{pdfTeX\sep PDF/X-1a\sep PDF/A-b}
\Title{Sample LaTeX input file}
\Author{LaTeX project team}
\Org{TeX Users Group}
```

You may note that the keywords are separated by `\sep` which will expand to XML elements `</rdf:li><rdf:li>` instead of comma character. This is the correct format required by the XMP metadata which is in XML format. Similarly, several other kinds of data can be captured using the following commands:

1. `\Subject` (this did not work in `pdfx`)
2. `\Creator`
3. `\Producer`
4. `\Volume`
5. `\Issue`

6. \CoverDisplayDate
7. \CoverDate
8. \Copyright
9. \Doi
10. \Lastpage
11. \Firstpage
12. \Journaltitle
13. \Journalnumber
14. \CreatorTool
15. \AuthoritativeDomain

The above commands are self-explanatory. Users can resort to alternate ways to create xmp file for inclusion in PDF. However, minimal \jobname.xmpdata shall be created with \Title and \Author commands along with their corresponding values for pdfx-ext package to work correctly. You may check Adobe XMP Development Center for more exhaustive information about Extensible Metadata Platform (XMP). An XMP Toolkit SDK which supports GNU/Linux, Macintosh and Windows operating systems is also provided under modified BSD licence.

pdfx-ext makes use of xmpincl package to include xmp data into the PDF. A good look at the documentation of xmpincl package will greatly help the users to understand the process of xmp data inclusion.

2.2. Limitations and dependencies

pdfx-ext.sty works only with PDFTEX. It further depends on the following packages:

1. xmpincl for insertion of metadata into PDF.
2. hyperref for hyperlinking, bookmarks, etc.
3. glyphtounicode.tex maps glyph names to corresponding Unicode.
4. glyphtounicode-cmr.tex does the same for cmr fonts.

Necessary color profile files may be obtained from the International Color Consortium. Please take a look at http://www.color.org/iccprofile.xalter.

2.3. Files included

Following files are included in the archive:

1. pdfx-ext.dtx — composite package and documentation.
2. pdfx-ext.ins — installer batch file.
3. pdfx-1a.xmp — specimen xmp template for PDF/X-1a.
4. pdfa-1b.xmp — specimen xmp template for PDF/A-1b.
5. small2e.xmpdata — specimen data file to provide values relating to the document to generate metadata.
6. glyphtounicode-cmr.tex — glyph names in cmr font to corresponding Unicode.

A directory named pdfx-ext may be created under $TEXMF/tex/latex and all *.sty, *.xmp and glyphtounicode-cmr.tex may be moved to the same. TEX's file database should then be updated by a suitable command depending on your distribution and operating system.

2.4. Options

The package can be loaded with the command:

```
\usepackage[<option>]{pdfx-ext}
```

where the options are:

`x-1a` generates PDF/X-1a compliant PDF.

`a-1b` generates PDF/A-1b compliant PDF.

2.5. Useful production notes

We have included some useful notes about production problems which we have encountered while generating PDF/A-1b compliant documents and the fixes recommended at: http://support.river-valley.com/wiki/index.php?title=Generating_PDF/A_compliant_PDFs_from_pdftex.

2.6. Miscellaneous information

The package is released under the LaTeX Project Public Licence. Bug reports, suggestions, feature requests, etc., may be sent to the authors at cvr@river-valley.org and/or thanh@river-valley.org. ben@companjen.name

There is a GitHub repository for this package: https://github.com/bencomp/pdfx-ext

3. Implementation

3.1. Various auxiliary macros

Two booleans are defined to switch between two options, `a-1b` and `x-1a`. PDF/X-1a further demands PDF version 1.3 and properly placed mediabox, bleedbox and trimbox (innermost) in that order. The MediaBox defines the size of the entire document, either the ArtBox or the TrimBox, defines the extent of the printable area. If the file is to be printed with bleed, a BleedBox, which must be larger than the TrimBox/ArtBox, but smaller than the MediaBox, must be defined.

```
⟨*package⟩
%
% $Id: pdfx-ext.dtx,v 1.00 2012/06/12 21:36:00 bc Exp bc $
%
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{pdfx-ext}
  [2012/06/12 v1.00 Initial update since pdfx (BC)]

\newif\ifpdfxonea \pdfxoneafalse
\newif\ifpdfaoneb \pdfaonebfalse

\DeclareOption{a-1b}{\global\pdfaonebtrue}
\DeclareOption{x-1a}{\global\pdfxoneatrue}
\ProcessOptions

\ifpdfxonea
 \pdfminorversion=3
 \pdfpageattr{/MediaBox[0 0 595 793]
```

```
              /BleedBox[0 0 595 793]
              /TrimBox[25 20 570 773]}
\else
 \pdfminorversion=4
\fi
```

Several macros were defined to capture data for the XMP metadata to be inserted into the PDF during generation.

```
\def\hash{\expandafter\@gobble\string\#}
\def\amp{\expandafter\@gobble\string\&}
\def\xmpAmp{\amp\hash x0026;}
\def\sep{</rdf:li><rdf:li>}
\def\TextCopyright{\amp\hash x00A9;}
\def\Title#1{\gdef\xmpTitle{#1}}
 \let\xmpTitle\@empty
\def\Author#1{\gdef\xmpAuthor{#1}}
 \let\xmpAuthor\@empty
\def\AuthorURI#1{\gdef\xmpAuthorURI{#1}}
 \let\xmpAuthorURI\@empty
\def\Subject#1{\gdef\xmpSubject{#1}}
 \let\xmpSubject\@empty
\def\Keywords#1{\gdef\xmpKeywords{#1}}
 \let\xmpKeywords\@empty
\def\Creator#1{\gdef\xmpCreator{#1}}
 \def\xmpCreator{\@pdfcreator}
\def\Producer#1{\gdef\xmpProducer{#1}}
 \def\xmpProducer{pdfTeX}
\def\Volume#1{\gdef\xmpVolume{#1}}
 \let\xmpVolume\@empty
\def\Issue#1{\gdef\xmpIssue{#1}}
 \let\xmpIssue\@empty
\def\CoverDisplayDate#1{\gdef\xmpCoverDisplayDate{#1}}
 \let\xmpCoverDisplayDate\@empty
\def\CoverDate#1{\gdef\xmpCoverDate{#1}}
 \let\xmpCoverDate\@empty
\def\Copyright#1{\gdef\xmpCopyright{#1}}
 \let\xmpCopyright\@empty
\def\Doi#1{\gdef\xmpDoi{#1}}
 \let\xmpDoi\@empty
\def\Lastpage#1{\gdef\xmpLastpage{#1}}
 \let\xmpLastpage\@empty
\def\Firstpage#1{\gdef\xmpFirstpage{#1}}
 \let\xmpFirstpage\@empty
\def\Journaltitle#1{\gdef\xmpJournaltitle{#1}}
 \let\xmpJournaltitle\@empty
\def\Journalnumber#1{\gdef\xmpJournalnumber{#1}}
 \let\xmpJournalnumber\@empty
\def\Org#1{\gdef\xmpOrg{#1}}
 \let\xmpOrg\@empty
\def\CreatorTool#1{\gdef\xmpCreatorTool{#1}}
 \def\xmpCreatorTool{\xmpProducer}
\def\AuthoritativeDomain#1{\gdef\xmpAuthoritativeDomain{#1}}
 \let\xmpAuthoritativeDomain\@empty
```

3.2. Document and instance ID's

\findUUID
\uuid

Document ID and instance ID are created from values obtained from `\jobname.pdf` and `\pdfcreationdate` by making use of `\pdfmdfivesum` primitive of PDFTEX.

```
\def\findUUID#1{\edef\tmpstring{\pdfmdfivesum{#1}}
    \expandafter\eightofnine\tmpstring\end}
\def\eightofnine#1#2#3#4#5#6#7#8#9\end{%
    \xdef\eightchars{#1#2#3#4#5#6#7#8}
    \fouroffive#9\end}
\def\fouroffive#1#2#3#4#5\end{\xdef\ffourchars{#1#2#3#4}
    \sfouroffive#5\end}
\def\sfouroffive#1#2#3#4#5\end{\xdef\sfourchars{#1#2#3#4}
    \tfouroffive#5\end}
\def\tfouroffive#1#2#3#4#5\end{\xdef\tfourchars{#1#2#3#4}
    \xdef\laststring{#5}}

\def\uuid{\eightchars-%
          \ffourchars-%
          \sfourchars-%
          \tfourchars-%
          \laststring}

\findUUID{\jobname.pdf}
\edef\xmpdocid{\uuid}
\findUUID{\pdfcreationdate}
\edef\xmpinstid{\uuid}
```

`\jobname.xmpdata` is read if available and the package, `xmpincl` is also loaded which will take care of inserting metadata into the PDF document.

```
\InputIfFileExists{\jobname.xmpdata}{}{}
\RequirePackage{xmpincl}
```

\convDate
\convertDate

The date format needed by metadata is different from the value provided by the `\pdfcreationdate`. `\convertDate` macro generates the required date format from `\pdfcreationdate`.

```
\def\convertDate{\getYear}
{\catcode`\D=12
 \gdef\getYear D:#1#2#3#4{\edef\xYear{#1#2#3#4}\getMonth}
}
\def\getMonth#1#2{\edef\xMonth{#1#2}\getDay}
\def\getDay#1#2{\edef\xDay{#1#2}\getHour}
\def\getHour#1#2{\edef\xHour{#1#2}\getMin}
\def\getMin#1#2{\edef\xMin{#1#2}\getSec}
\def\getSec#1#2{\edef\xSec{#1#2}\getTZh}
\def\getTZh +#1#2{\edef\xTZh{#1#2}\getTZm}
\def\getTZm '#1#2'{%
    \edef\xTZm{#1#2}%
    \edef\convDate{\xYear-\xMonth-\xDay
      T\xHour:\xMin:\xSec+\xTZh:\xTZm}}
\expandafter\convertDate\pdfcreationdate
```

3.3. Color profiles

/OutputIntents

For better color management, PDF/X-1a and PDF/A-1b need an ICC profile included in the document. An ICC profile is a set of data that characterizes a color input or output device, or a color space, according to standards promulgated by the International

Color Consortium (ICC). Profiles describe the color attributes of a particular device or viewing requirement by defining a mapping between the device source or target color space and a profile connection space. For PDF/X-1a, we have included the ICC profile namely, FOGRA39L.icc which is for CMYK data and for PDF/A-1b, we have used sRGBIEC1966-2.1.icm for RGB data. You can change the value of the file attribute in the code below to use different color profile files.

```
\ifpdfxonea
 \def\@pctchar{\expandafter\@gobble\string\%}
 \def\@bchar{\expandafter\@gobble\string\\}
 \immediate\pdfobj stream attr{/N 4}  file{FOGRA39L.icc}
 \edef\OBJ@CVR{\the\pdflastobj}
 \pdfcatalog{/OutputIntents [ <<
   /Type/OutputIntent
   /S/GTS_PDFX
   /OutputCondition (FOGRA39)
   /OutputConditionIdentifier (FOGRA39 \@bchar(ISO Coated v2
    300\@pctchar\space \@bchar(ECI\@bchar)\@bchar))
   /DestOutputProfile \OBJ@CVR\space 0 R
   /RegistryName(http://www.color.org)
  >> ]}
\else
 \immediate\pdfobj stream attr{/N 4}  file{sRGBIEC1966-2.1.icm}
 \edef\OBJ@RVT{\the\pdflastobj}
 \pdfcatalog{%
   /OutputIntents [ <<
   /Type /OutputIntent
   /S/GTS_PDFA1
   /DestOutputProfile \OBJ@RVT\space 0 R
   /OutputConditionIdentifier (sRGB IEC61966-2.1)
   /Info(sRGB IEC61966-2.1)
  >> ]}
\fi
```

One of the xmp files is selectively loaded based on the option chosen.

```
\begingroup
\let\&=\xmpAmp
\ifpdfxonea
 \includexmp{pdfx-1a}
\else
 \includexmp{pdfa-1b}
\fi
\endgroup
```

glyphtounicode.tex and glyphtounicode-cmr.tex are read. These files contain mapping from glyph names to corresponding unicode for embedded fonts, which are required by PDF/A-1b. glyphtounicode.tex covers AGL (Adobe Glyph List), names from texglyphlist.txt (part of lcdf-typetools) and zapfdingbats.txt, plus a few exceptions. glyphtounicode-cmr.tex covers glyphs that are used in CM fonts but not listed in glyphtounicode.txt; the mappings come from file goadb998.nam (part of TEX Gyre fonts).

```
\input glyphtounicode.tex
\input glyphtounicode-cmr.tex
\pdfgentounicode=1
```

Active content is not allowed in a PDF/X-1a file. This means that standard PDF features like forms, signatures, comments and embedded sounds and movies are not

allowed in PDF/X-1a. So hyperref is loaded in draft mode and an info dictionary is defined with \pdfinfo command.

```
\ifpdfxonea
  \RequirePackage[draft,pdftex,pdfpagemode=UseNone,bookmarks=false]{hyperref}
  \pdfinfo{
      /Title(\xmpTitle)%
      /Author(\xmpAuthor)%
      /Creator(\xmpProducer)%
      /CreationDate(\convDate)%
      /ModDate(\convDate)%
      /Producer(\xmpProducer)%
      /Trapped /False
      /GTS_PDFXVersion (PDF/X-1:2001)%
      /GTS_PDFXConformance (PDF/X-1a:2001)%
  }
\else
```

For PDF/A-1b, load hyperref package with pdfa option, so that it will take care of the link annotations correctly. We have slightly modified the \pdfinfo by including /GTS_PDFA1Version (PDA/A-1b:2005). Take a look at the modified \PDF@FinishDoc macro of hyperref.

```
  \RequirePackage[pdftex,pdfa]{hyperref}
\def\PDF@FinishDoc{%
  \Hy@UseMaketitleInfos
  \pdfinfo{%
     /Creator(\xmpProducer)%
     \ifx\@pdfcreationdate\@empty
    \else
      /CreationDate(\@pdfcreationdate)%
    \fi
    \ifx\@pdfmoddate\@empty
    \else
      /ModDate(\@pdfmoddate)%
    \fi
      /Producer(\xmpProducer)%
     /Trapped /False
     /GTS_PDFA1Version (PDF/A-1b:2005)%
  }%
  \Hy@DisableOption{pdfauthor}%
  \Hy@DisableOption{pdftitle}%
  \Hy@DisableOption{pdfsubject}%
  \Hy@DisableOption{pdfcreator}%
  \Hy@DisableOption{pdfcreationdate}%
  \Hy@DisableOption{pdfmoddate}%
  \Hy@DisableOption{pdfproducer}%
  \Hy@DisableOption{pdfkeywords}}
\fi
%
⟨/package⟩
```