The hyperxmp package*

Scott Pakin scott+hyxmp@pakin.org

September 24, 2014

Abstract

hyperxmp makes it easy for an author to include XMP metadata in a PDF document produced by LATEX. hyperxmp integrates seamlessly with hyperref and requires virtually no modifications to a document that already specifies document metadata through hyperref's mechanisms.

1 Introduction

Adobe Systems, Inc. has been promoting XMP [4]—eXtensible Metadata Platform—as a standard way to include metadata within a document. The idea behind XMP is that it is an XML-based description of various document attributes and is embedded as uncompressed, unencoded text within the document it describes. By storing the metadata this way it is independent of the document's file format. That is, regardless of whether a document is in PDF, JPEG, HTML, or any other format, it is trivial for a program (or human) to locate, extract, and—using any standard XML parser—process the embedded XMP metadata.

As of this writing there are few tools that actually do process XMP. However, it is easy to imagine future support existing in file browsers for displaying not only a document's filename but also its title, list of authors, description, and other metadata.

This is too abstract! Give me an example. Consider a LATEX document with three authors: Jack Napier, Edward Nigma, and Harvey Dent. The generated PDF file will contain, among other information, the following stanza of XMP code embedded within it:

```
<dc:creator>
  <rdf:Seq>
   <rdf:li>Jack Napier</rdf:li>
   <rdf:li>Edward Nigma</rdf:li>
  <rdf:li>Harvey Dent</rdf:li>
```

^{*}This document corresponds to hyperxmp v2.6, dated 2014/09/24.

```
</rdf:Seq>
</dc:creator>
```

In the preceding code, the dc namespace refers to the Dublin Core schema, a collection of metadata properties. The dc:creator property surrounds the list of authors. The rdf namespace is the Resource Description Framework, which defines rdf:Seq as an ordered list of values. Each author is represented by an individual list item (rdf:li), making it easy for an XML parser to separate the authors' names.

Remember that XMP code is stored as *metadata*. It does not appear when viewing or printing the PDF file. Rather, it is intended to make it easy for applications to identify and categorize the document.

What metadata does hyperxmp process? hyperxmp knows how to embed all of the following types of metadata within a document:

- authors (dc:creator)
- base URL (xmp:BaseURL)
- contact address (Iptc4xmpCore:CreatorContactInfo/CiAdrExtadr, Iptc4xmpCore:CreatorContactInfo/CiAdrCity, Iptc4xmpCore:CreatorContactInfo/CiAdrRegion, Iptc4xmpCore:CreatorContactInfo/CiAdrPcode, and Iptc4xmpCore:CreatorContactInfo/CiAdrCtry)
- contact email address(es) (Iptc4xmpCore:CreatorContactInfo/CiEmailWork)
- contact telephone number(s) (Iptc4xmpCore:CreatorContactInfo/CiTelWork)
- contact URL(s) (lptc4xmpCore:CreatorContactInfo/CiUrlWork)
- copyright (dc:rights and xmpRights:Marked)
- date (dc:date, xmp:CreateDate, xmp:ModifyDate, and xmp:MetadataDate)
- document identifier (xmpMM:DocumentID)
- document instance identifier (xmpMM:InstanceID)
- file format (dc:format)
- keywords (pdf:Keywords and dc:subject)
- language (dc:language)
- LATEX file name (dc:source)
- license URL (xmpRights:WebStatement)
- metadata writer (photoshop:CaptionWriter)

- PDF version (pdf:PDFVersion)
- PDF-generating tool (pdf:Producer and xmp:CreatorTool)
- PDF/A compliance level and version (pdfaid:part and pdfaid:conformance)
- primary author's position/title (photoshop:AuthorsPosition)
- summary (dc:description)
- title (dc:title)

More types of metadata may be added in a future release.

How does hyperxmp compare to the xmpincl package? The short answer is that xmpincl is more flexible but hyperxmp is easier to use. With xmpincl, the author manually constructs a file of arbitrary XMP data and the package merely embeds it within the generated PDF file. With hyperxmp, the author specifies values for various predefined metadata types and the package formats those values as XMP and embeds the result within the generated PDF file.

xmpincl can embed XMP only when running under pdfIATEX and only when in PDF-generating mode. hyperxmp additionally works with a few other PDF-producing IATEX backends.

hyperxmp and xmpincl can complement each other. An author may want to use hyperxmp to produce a basic set of XMP code, then extract the XMP code from the PDF file with a text editor, augment the XMP code with any metadata not supported by hyperxmp, and use xmpincl to include the modified XMP code in the PDF file.

2 Usage

hyperxmp works by postprocessing some of the package options honored by hyperref. To use hyperxmp, merely put a \usepackage{hyperxmp} in your document's preamble. That line can appear anywhere before the hyperref PDF options are specified (i.e., with either \usepackage[...]{hyperref} or \hypersetup{...}). hyperxmp will construct its XMP data using the following hyperref options:

- baseurl
- pdfauthor
- pdfkeywords
- pdflang
- pdfproducer
- pdfsubject

• pdftitle

hyperxmp instructs hyperref also to accept the following options, which have meaning only to hyperxmp:

- pdfauthortitle
- pdfcaptionwriter
- pdfcontactaddress
- pdfcontactcity
- pdfcontactcountry
- pdfcontactemail
- pdfcontactphone
- pdfcontactpostcode
- pdfcontactregion
- pdfcontacturl
- pdfcopyright
- pdfdate
- pdflicenseurl
- pdfmetalang

pdfauthortitle indicates the primary author's position or title. pdfcaptionwriter specifies the name of the person who added the metadata to the document. The next eight items describe how to contact the person or institution responsible for the document (the "contact"). pdfcontactaddress is the contact's street address and can include the institution name if the contact is an institution; pdfcontactcity is the contact's city. pdfcontactcountry is the contact's country; pdfcontactemail is the contact's email address (or multiple, comma-separated email addresses); pdfcontactphone is the contact's telephone number (or multiple, comma-separated telephone numbers); pdfcontactpostcode is the contact's postal code; pdfcontactregion is the contact's state or province; and pdfcontacturl is the contact's URL (or multiple, comma-separated URLs).

pdfcopyright defines the copyright text. pdfdate specifies the document date. It is analogous to the LATEX \date command, and, like \date, defaults to the date the document was built. However, pdfdate must be specified in YYYY-MM-DDThh:mm:ss.ff+TT:tt format as per the W3C's recommendation [11]. For example, 14 hours, 15 minutes, 9.26 seconds past midnight U.S. Mountain Daylight Time (UTC-6) on the 23rd day of September in the year 2014 should be written as 2014-09-23T14:15:09.26-06:00. This can be truncated

to 2014-09-23T14:15:09-06:00 or 2014-09-23T14:15-06:00 or 2014-09-23 or 2014-09 or 2014 but no other subsets. hyperxmp does not validate pdfdate's argument, but an invalid format may confuse a PDF reader.

pdflicenseurl identifies a URL that points to the document's license agreement. pdfmetalang indicates the natural language in which the metadata is written, typically as an IETF language tag [7], for example, "en" for English, "en-US" for specifically United States English, "de" for German, and so forth. If pdfmetalang is not specified, hyperxmp assumes the metadata language is the same as the document language (hyperref's pdflang option). If neither pdfmetalang nor pdflang is specified, hyperxmp uses only "x-default" as the metadata language. Note that "x-default" metadata is always included in addition to the specified metadata language, as the user reading the document may not have specified a language preference.

It is usually more convenient to provide values for those options using hyperref's \hypersetup command than on the \usepackage command line. See the hyperref manual for more information. The following is a sample IATEX document that provides values for most of the metadata options that hyperxmp recognizes:

```
\documentclass{article}
\usepackage{hyperxmp}
\usepackage{hyperref}
\title{%
 On a heuristic viewpoint concerning the production and
 transformation of light}
\author{Albert Einstein}
\date{March 17, 1905}
\hypersetup{%
 pdftitle={%
   On a heuristic viewpoint concerning the production and
    transformation of light},
 pdfauthor={Albert Einstein},
 pdfauthortitle={Technical Assistant, Level III},
 pdfdate={1905-03-17},
 pdfcopyright={Copyright (C) 1905, Albert Einstein},
 pdfsubject={photoelectric effect},
 pdfkeywords={energy quanta, Hertz effect, quantum physics},
 pdflicenseurl={http://creativecommons.org/licenses/by-nc-nd/3.0/},
 pdfcaptionwriter={Scott Pakin},
 pdfcontactaddress={Kramgasse 49},
 pdfcontactcity={Bern},
 pdfcontactpostcode={3011},
 pdfcontactcountry={Switzerland},
 pdfcontactphone={031 312 00 91},
 pdfcontactemail={aeinstein@ipi.ch},
 pdfcontacturl={%
   http://einstein.biz/,
   https://www.facebook.com/AlbertEinstein
 pdflang={en},
```

```
baseurl={http://www.ctan.org/tex-archive/macros/latex/contrib/hyperxmp/}
}
\begin{document}
\maketitle
A profound formal difference exists between the theoretical
concepts that physicists have formed about gases and other
ponderable bodies, and Maxwell's theory of electromagnetic
processes in so-called empty space\dots
\end{document}
```

Compile the document to PDF using any of the following approaches:

- pdfIATEX
- LuaLATEX
- LATEX + Dvipdfm
- LATEX + Dvips + Adobe Acrobat Distiller
- Xalatex

Unfortunately, the IATEX + Dvips + Ghostscript path doesn't work. Ghostscript bug report #690066, closed with "WONTFIX" status on 2012-05-28, explains that Ghostscript doesn't honor the Metadata tag needed to inject a custom XMP packet. Instead, Ghostscript fabricates an XMP packet of its own based on the metadata it finds in the PDF file's Info dictionary (Author, Title, Subject, and Keywords).

Once the document is compiled, the resulting PDF file will contain an XMP packet that looks something like that shown in Appendix A. Figure 1 is a screenshot of the XMP metadata as it appears in Adobe Acrobat's "Advanced" metadata dialog box. Further clicking on the "Advanced" item within that dialog box displays all of the document's metadata sorted by schema as shown in Figure 2.

Note 1: Acrobat Author bug A bug in Adobe Acrobat—at least in versions 10.0.1 and earlier—causes that PDF reader to confuse the XMP and non-XMP author lists when displaying the document's metadata. Specifically, the first author is displayed as the concatenated list of authors from the non-XMP data (Author) while the remaining authors are displayed from the XMP data (dc:creator). For example, suppose that a document's authors are Jack Napier, Edward Nigma, and Harvey Dent. When displaying the document properties, Adobe Acrobat replaces "Jack Napier" with a single author named "Jack Napier, Edward Nigma, Harvey Dent" and leaves "Edward Nigma" and "Harvey Dent" as the second and third authors, respectively.

\XMPTruncateList

The hyperxmp package provides a workaround for this bug in the form of the \XMPTruncateList macro. \XMPTruncateList takes the name of a list (a hyperref

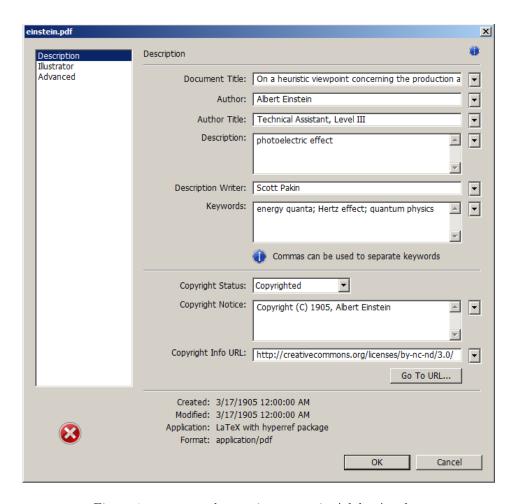


Figure 1: XMP metadata as it appears in Adobe Acrobat

option name) and replaces the list with the value of its first element. Currently, the only meaningful usage is to put

\XMPTruncateList{pdfauthor}

in your document's preamble. This will cause Adobe Acrobat to properly display all of the authors but at the cost of other PDF readers likely displaying only the first author.

Note 2: Acrobat multiline-field bug The IPTC Photo Metadata schema states that "the [contact] address is a multiline field" [6]. hyperxmp converts commas in pdfcontactaddress's argument to line breaks in the generated XML.

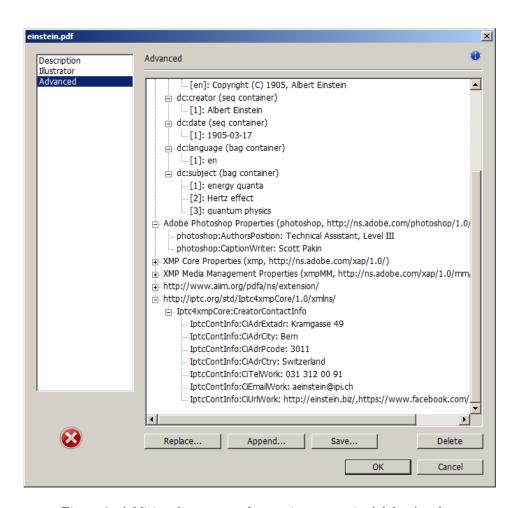


Figure 2: Additional XMP metadata as it appears in Adobe Acrobat

\xmplinesep

Unfortunately, A bug in Adobe Acrobat—at least in versions 10.0.1 and earlier—causes that PDF reader to discard line breaks in the contact address. Interestingly, Adobe Illustrator CS5 correctly displays the contact address. If you find Adobe Acrobat's behavior bothersome, you can redefine the \mmplinesep macro as a string to use as an address-line separator. For example, the following replaces all commas appearing in pdfcontactaddress's argument with semicolons:

\renewcommand*{\xmlinesep}{;}

Note 3: XALATEX object compression XALATEX (or, more precisely, the xdvipdfmx back end), compresses all PDF objects, including the ones containing XMP metadata. While Adobe Acrobat can still detect and utilize the XMP metadata, non-PDF-aware applications are unlikely to see the metadata. Three

options to consider are to (1) use a different program (e.g., LuaIATEX), (2) pass the --output-driver="xdvipdfmx -z0" option to XTIATEX to instruct xdvipdfmx to turn off all compression (which will of course make the PDF file substantially larger), or (3) postprocess the generated PDF file by loading it into the commercial version of Adobe Acrobat and re-saving it with the Save As... menu option.

Note 4: Literal commas hyperxmp splits the pdfauthor and pdfkeywords lists at commas. Therefore, when specifying pdfauthor and pdfkeywords, you should separate items with commas. Also, omit "and" and other text that does not belong to any list item. The following examples should serve as clarification:

```
\mathbf{Wrong:} \ \mathtt{pdfauthor=\{Jack\ Napier,\ Edward\ Nigma,\ and\ Harvey\ Dent\}}
```

Wrong: pdfauthor={Jack Napier; Edward Nigma; Harvey Dent}

Right: pdfauthor={Jack Napier, Edward Nigma, Harvey Dent}

\xmpcomma \xmpquote

If you need to include a literal comma within an author or keyword list (where commas normally separate list items) or a street address (where commas normally separate lines), use the \mathbb{xmpcomma} macro to represent it, and wrap the entire entry containing the comma within \mathbb{xmpquote}\{\ldots\} as shown below:

As of version 2.2 of hyperxmp, it is acceptable to use \mpcomma and \mpquote within any hyperxmp option, not just in those in which a comma normally serves as a separator (i.e., lists and multiline fields). Outside of cases in which a comma serves as a separator, \mpcomma is treated as an ordinary comma, and \mpquote returns its argument unmodified. Hence, it is legitimate to use \mpcomma and \mpquote in cases like the following

pdfauthortitle={\xmpquote{Psychiatrist\xmpcomma\ Arkham Asylum}}

(Like most hyperxmp options, pdfauthortitle inserts its argument unmodified in an XMP tag.) When in doubt, use \mathbb{xmpcomma} and \mathbb{xmpquote}; it should always be safe to do so.

\xmptilde

Version 2.4 of hyperxmp introduces a convenience macro called \xmptilde.\xmptilde expands to a literal tilde character instead of the nonbreaking space that "~" normally represents. Use it to represent URLs such as http://www.pakin.org/~scott/ ("http://www.pakin.org/\xmptilde scott/") in options such as baseurl, pdfcontacturl and pdflicenseurl.

Note 5: Unicode support Unicode support is provided via the hyperref package. If you specify unicode=true either as a hyperref option or as an argument to the \hypersetup command, the document can include Unicode characters in its XMP fields.

3 Implementation

This section presents the commented LATEX source code for hyperxmp. Read this section only if you want to learn how hyperxmp is implemented.

3.1 Initial preparation

\hyxmp@dq@code

The ngerman package redefines "" as an active character, which causes problems for hyperxmp when it tries to use that character. We therefore save the double-quote character's current category code in \hyxmp@dq@code and mark the character as category code 12 ("other"). The original category code is restored at the end of the package code (Section 3.7).

- 1 \edef\hyxmp@dq@code{\the\catcode'\"}
- 2 \catcode '\"=12

\hyxmp@at@end \hyxmp@driver

The \hyxmp@at@end macro includes code at the end of the document. For pdfTEX, the standard \AtEndDocument works well enough. For all the other backends we use \AtEndDvi from the atenddvi package, which is more robust but requires an addition LATEX run.

- 3 \def\hyxmp@driver{hpdftex}
- 5 \let\hyxmp@at@end=\AtEndDocument
- 6 \else
- 7 \RequirePackage{atenddvi}
- 8 \let\hyxmp@at@end=\AtEndDvi
- 9 \fi

3.2 Integration with hyperref

An important design decision underlying hyperxmp is that the package should integrate seamlessly with hyperref. To that end, hyperxmp takes its XMP metadata from hyperref's baseurl, pdfauthor, pdfkeywords, pdflang, pdfproducer, pdfsubject, and pdftitle options. It also introduces a number of new options, which are listed on page 4. For consistency with hyperref's document-metadata naming conventions (which are in turn based on IATEX's document-metadata naming conventions), we do not prefix metadata-related macro names with our package-specific \hyxmp@ prefix. That is, we use names like \@pdfcopyright instead of \hyxmp@pdfcopyright.

We load a bunch of helper packages: kvoptions for package-option processing, pdfescape and stringenc for re-encoding Unicode strings, intcalc for performing integer calculations (division and modulo), and ifxetex for detecting X₇T_FX.

```
10 \RequirePackage{kvoptions}
11 \RequirePackage{pdfescape}
12 \RequirePackage{stringenc}
```

13 \RequirePackage{intcalc} 14 \RequirePackage{ifxetex}

\hyxmp@pdfstringdef \hyxmp@textunderscore

Because hyperxmp uses underscores to represent hard spaces, we need "_" to map initially to something other than an underscore, in particular the ASCII NAK (^~U) character. To accomplish this, we wrap hyperref's \pdfstringdef macro with our own version that temporarily does the proper substitution. Later in the execution, after underscores have been replaced with spaces, we replace NAK characters with underscores.

```
15 \newcommand{\hyxmp@pdfstringdef}[2]{%
    \let\hyxmp@textunderscore=\textunderscore
    \let\textunderscore=\hyxmp@uscore
    \pdfstringdef{#1}{#2}%
18
   \let\textunderscore=\hyxmp@textunderscore
19
20 }
```

\Opdfdatetime Prepare to store the document's date and (optionally) time.

```
21 \def\@pdfdatetime{}
```

\Opdfcopyright Prepare to store the document's copyright statement.

```
23 \def\@pdfcopyright{}
```

24 \define@key{Hyp}{pdfcopyright}{\hyxmp@pdfstringdef\@pdfcopyright{#1}}

\@pdflicenseurl Prepare to store the URL containing the document's license agreement.

```
25 \def\@pdflicenseurl{}
```

 $26 \end{fine} \end{fierare} \end{fine} \end{fine} \end{fine} \end{fine} \end{fine} \en$

\Opdfauthortitle Prepare to store the author's position/title (e.g., Staff Writer).

```
27 \def\@pdfauthortitle{}
```

28 \define@key{Hyp}{pdfauthortitle}{\hyxmp@pdfstringdef\@pdfauthortitle{#1}}

\@pdfcaptionwriter Prepare to store the name of the person who inserted the hyperxmp metadata.

```
29 \def\@pdfcaptionwriter{}
```

30 \define@key{Hyp}{pdfcaptionwriter}{\hyxmp@pdfstringdef\@pdfcaptionwriter{#1}}

\Copdfmetalang Prepare to store the natural language of the document's metadata, typically as an ISO 639-1 two-letter abbreviation.

```
31 \def\@pdfmetalang{}
```

32 \define@key{Hyp}{pdfmetalang}{\hyxmp@pdfstringdef\@pdfmetalang{#1}}

The following eight macros—\Opdfcontactaddress, \Opdfcontactcity, \@pdfcontactpostcode, \@pdfcontactregion, \@pdfcontactcountry, \@pdfcontactphone, \@pdfcontactemail, and \@pdfcontacturl—together specify how to contact the person or institution responsible for the document.

\@pdfcontactaddress

Prepare to store a street address for the document's contact person/institution. The IPTC standard defines this as follows:

The contact information address part. Comprises an optional company name and all required information to locate the building or postbox to which mail should be sent. To that end, the address is a multiline field.

For consistency with the rest of hyperxmp, we use commas to separate terms, in this case, lines of the address. The author can use \xmpquote and \xmpcomma to include literal commas.

```
33 \def\@pdfcontactaddress{}
34 \define@key{Hyp}{pdfcontactaddress}{%
    \let\xmpcomma=\hyxmp@comma
   \def\xmpquote##1{##1}%
36
   \hyxmp@pdfstringdef\@pdfcontactaddress{#1}%
37
   \def\xmpcomma{,}%
39
   \let\xmpquote=\relax
40 }
```

\@pdfcontactcity

Prepare to store the city of the document's contact person/institution.

- 41 \def\@pdfcontactcity{}

\@pdfcontactregion Prepare to store the state or province of the document's contact person/institution.

- 43 \def\@pdfcontactregion{}
- 44 \define@key{Hyp}{pdfcontactregion}{\hyxmp@pdfstringdef\@pdfcontactregion{#1}}

\@pdfcontactpostcode

Prepare to store the postal code of the document's contact person/institution.

- 45 \def\@pdfcontactpostcode{}
- 46 \define@key{Hyp}{pdfcontactpostcode}{\hyxmp@pdfstringdef\@pdfcontactpostcode{#1}}

\@pdfcontactcountry

Prepare to store the country of the document's contact person/institution.

- 47 \def\@pdfcontactcountry{}
- 48 \define@key{Hyp}{pdfcontactcountry}{\hyxmp@pdfstringdef\@pdfcontactcountry{#1}}

\@pdfcontactphone Prepare to store the telephone number of the document's contact person/institution.

- 49 \def\@pdfcontactphone{}
- 50 \define@key{Hyp}{pdfcontactphone}{\hyxmp@pdfstringdef\@pdfcontactphone{#1}}

\@pdfcontactemail Prepare to store the email address of the document's contact person/institution.

- 51 \def\@pdfcontactemail{}
- 52 \define@key{Hyp}{pdfcontactemail}{\hyxmp@pdfstringdef\@pdfcontactemail{#1}}

\@pdfcontacturl Prepare to store the URL of the document's contact person/institution.

- 53 \def\@pdfcontacturl{}
- 54 \define@key{Hyp}{pdfcontacturl}{\hyxmp@pdfstringdef\@pdfcontacturl{#1}}

We need to capture list arguments (viz. pdfauthor and pdfkeywords) before hyperref converts them to PDFDocEncoding. Otherwise, \mmpcomma is permanently replaced with a comma, and we lose our ability to change it to a \hyxmp@comma. We therefore need to augment hyperref's option processing with our own. Because hyperref has not yet been loaded we need to ensure that our augmentation gets loaded in the future: after the \usepackage{hyperref} but before options are passed to that package.

For lack of a better approach, hyperxmp redefines \ProcessKeyvalOptions to alter the way hyperref processes pdfauthor and pdfkeywords. This is somewhat heavy-handed as it gets executed for every subsequently loaded package that uses \ProcessKeyvalOptions, but at least it does what we need. hyperxmp also redefines \hypersetup to do the same thing. This is required in case hyperref is loaded before hyperxmp.

\hyxmp@pdfauthor \hyxmp@pdfkeywords

Prepare to store the name of the author and a list of keywords.

- 55 \def\hyxmp@pdfauthor{}
- 56 \def\hyxmp@pdfkeywords{}

\hyxmp@redefine@Hyp

If not already redefined, redefine hyperref's pdfauthor and pdfkeywords options to properly handle \xmpcomma and \xmpquote.

57 \newcommand*{\hyxmp@redefine@Hyp}{%

\hyxmp@Hyp@pdfauthor

Store the old definition of \KV@Hyp@pdfauthor in \hyxmp@Hyp@pdfauthor, but only if we see that \KV@Hyp@pdfauthor is defined and \hyxmp@Hyp@pdfauthor isn't. Otherwise, we'd be defining \hyxmp@Hyp@pdfauthor in terms of itself and creating an infinite loop.

```
\@ifundefined{KV@Hyp@pdfauthor}{}{%
58
      \@ifundefined{hyxmp@Hyp@pdfauthor}{%
59
        \expandafter\let\expandafter\hyxmp@Hyp@pdfauthor
60
           \csname KV@Hyp@pdfauthor\endcsname
61
      }{}%
62
    }%
```

\KV@Hyp@pdfauthor \xmpcomma \xmpquote \hyxmp@pdfauthor \@pdfauthor

Redefine \KV@Hyp@pdfauthor to process its argument twice. The first time, \mmpcomma is defined as a placeholder character (\hymmp@comma) and \mmpquote as the identity function. The result is stored in \hyxmp@pdfauthor for use in structured lists (those surrounding each entry with <rdf:li>). The second time, \mmpcomma is defined as an ordinary comma, and \mmpquote is defined as a macro that puts its argument within double quotes. The result is stored in \Opdfauthor for use in unstructured lists (those in which the entire list appears within a single pair of tags).

```
\define@key{Hyp}{pdfauthor}{%
64
```

\let\xmpcomma=\hyxmp@comma 65

```
\def\xmpquote###1{###1}%
66
      \hyxmp@Hyp@pdfauthor{##1}%
67
      \global\let\hyxmp@pdfauthor=\@pdfauthor
68
      \def\xmpcomma{,}%
69
      \def\xmpquote###1{"####1"}%
70
71
      \hyxmp@Hyp@pdfauthor{##1}%
72
      \def\xmpcomma{,}%
      \let\xmpquote=\relax
73
    }%
74
```

\hyxmp@Hyp@pdfkeywords

The previous block of code now repeats for the keyword list, starting by storing the old definition of \KV@Hyp@pdfkeywords in \hyxmp@Hyp@pdfkeywords.

```
75 \@ifundefined{KV@Hyp@pdfkeywords}{}\%
76 \@ifundefined{hyxmp@Hyp@pdfkeywords}{\%
77 \expandafter\let\expandafter\hyxmp@Hyp@pdfkeywords
78 \csname KV@Hyp@pdfkeywords\endcsname
79 \}{\}\%
80 \}\%
```

\KV@Hyp@pdfkeywords \xmpcomma \xmpquote \hyxmp@pdfkeywords \@pdfkeywords Redefine \KV@Hyp@pdfkeywords to process its argument twice. The first time, \mpcomma is defined as a placeholder character (\hymp@comma) and \mpquote as the identity function. The result is stored in \hymp@pdfkeywords for use in structured lists (those surrounding each entry with <rdf:li>). The second time, \mpcomma is defined as an ordinary comma, and \mpquote is defined as a macro that puts its argument within double quotes. The result is stored in \@pdfkeywords for use in unstructured lists (those in which the entire list appears within a single pair of tags).

```
\define@key{Hyp}{pdfkeywords}{%
81
82
      \let\xmpcomma=\hyxmp@comma
      \def\xmpquote###1{###1}%
83
      \hyxmp@Hyp@pdfkeywords{##1}%
84
      \global\let\hyxmp@pdfkeywords=\@pdfkeywords
85
      \def\xmpcomma{,}%
86
      \def\xmpquote###1{"####1"}%
87
      \hyxmp@Hyp@pdfkeywords{##1}%
88
89
      \def\xmpcomma{,}%
      \let\xmpquote=\relax
90
    }%
91
92 }
```

\hyxmp@ProcessKeyvalOptions \ProcessKeyvalOptions Redefine kvoptions's \ProcessOptions command to invoke \hyxmp@redefine@Hyp before performing its normal option processing.

```
93 \let\hyxmp@ProcessKeyvalOptions=\ProcessKeyvalOptions
94 \renewcommand*{\ProcessKeyvalOptions}{%}
95 \hyxmp@redefine@Hyp
96 \hyxmp@ProcessKeyvalOptions
97 }
```

\hyxmp@hypersetup \hypersetup

Redefine hyperref's \hypersetup command to invoke \hyxmp@redefine@Hyp before performing its normal option processing.

```
98 \let\hyxmp@hypersetup=\hypersetup
99 \def\hypersetup{%
100 \hyxmp@redefine@Hyp
101 \hyxmp@hypersetup
102 }
```

\hyxmp@find@metadata \hyxmp@concated@metadata Issue a warning message if the author failed to specify any metadata at all. This excludes metadata that is included automatically such as the current timestamp. Note that we don't consider \@pdfmetalang as metadata as that value is meaningful only when used in conjunction with other information.

```
103 \newcommand*{\hyxmp@find@metadata}{%
     \edef\hyxmp@concated@metadata{%
       \@baseurl
105
       \@pdfauthor
106
       \@pdfauthortitle
107
       \@pdfcaptionwriter
108
       \@pdfcontactaddress
109
       \@pdfcontactcity
110
       \@pdfcontactcountry
111
       \@pdfcontactemail
112
       \@pdfcontactphone
113
       \@pdfcontactpostcode
114
       \@pdfcontactregion
115
       \@pdfcontacturl
116
       \@pdfcopyright
117
       \@pdfdatetime
118
       \@pdfkeywords
119
       \@pdflang
120
       \@pdflicenseurl
121
       \@pdfsubject
122
       \@pdftitle
123
124
     \ifx\hyxmp@concated@metadata\@empty
125
       \PackageWarningNoLine{hyperxmp}{%
126
127 \jobname.tex did not specify any metadata to\MessageBreak
128 include in the XMP packet.\space\space Please see the\MessageBreak
129 hyperxmp documentation for instructions on how to\MessageBreak
130 provide metadata values to hyperxmp}%
131
     \fi
132 }
```

Rather than load hyperref ourself we let the author do it then verify he actually did. This approach gives the author the flexibility to load hyperxmp and hyperref in either order and to call \hypersetup anywhere in the document's preamble, not just before hyperxmp is loaded.

```
133 \AtBeginDocument{%
134 \@ifpackageloaded{hyperref}{%
```

In older versions of hyperref, \@pdflang is set to \@empty if pdflang is not specified. In newer versions of hyperref, \@pdflang is set to \relax if pdflang is not specified. The latter is a bit problematic for hyperxmp because it makes \@pdflang non-expandable, which causes a literal "\@pdflang" to be written as XMP metadata. To avoid that situation we redefine \@pdflang as \@empty if we see it set to \relax.

```
135 \ifx\@pdflang\relax
136 \let\@pdflang=\@empty
137 \fi
```

If the author explicitly specified the language to use for the document's metadata, we use that. If not, we use the document language, specified to hyperref with the pdflang option. If the author did not specify a language, we use x-default as the metadata language.

```
138 \ifx\@pdflang\@empty
139 \let\@pdfmetalang=\hyxmp@x@default
140 \else
141 \edef\@pdfmetalang{\@pdflang}\%
142 \fi
143 \hyxmp@xmlify\@pdfmetalang
```

If the author explicitly specified the document date, override the compilation timestamp with the specified date.

```
144 \ifx\@pdfdatetime\@empty
145 \else
146 \edef\hyxmp@today{\@pdfdatetime}%
147 \fi
```

We wait until the end of the document to construct the XMP packet and write it to the PDF document catalog. This gives the author ample opportunity to provide metadata to hyperref and thereby hyperxmp.

```
\hyxmp@at@end{%
148
149
            \hyxmp@find@metadata
            \hyxmp@embed@packet
150
         }%
151
       }%
152
       {\PackageWarningNoLine{hyperxmp}{%
154 \jobname.tex failed to include a\MessageBreak
155 \string\usepackage\string{hyperref\string}
156 in the preamble.\MessageBreak
157 Consequently, all hyperxmp functionality will be\MessageBreak
158 disabled}%
       }%
159
160 }
```

3.3 Manipulating author-supplied data

The author provides metadata information to hyperxmp via package options to hyperref or via hyperref's \hypersetup command. The functions in this section convert author-supplied lists (e.g., pdfkeywords={foo, bar, baz}) into IATEX

lists (e.g., \@elt {foo} \@elt {bar} \ delt {baz}) that can be more easily manipulated (Section 3.3.1); trim spaces off the ends of strings (Section 3.3.2); and, in Section 3.3.3, convert text to XML (e.g., from <scott+hyxmp@pakin.org> to <scott+hyxmp@pakin.org>).

3.3.1 List manipulation

We define a macro for converting a list of comma-separated elements (e.g., the list of PDF keywords) to a list of LATEX \@elt-separated elements.

\hyxmp@commas@to@list

Given a macro name (#1) and a comma-separated list (#2), define the macro name as the elements of the list, each preceded by **\Qelt**. (Executing the macro therefore applies \@elt to each element in turn.)

```
161 \newcommand*{\hyxmp@commas@to@list}[2]{%
162
     \gdef#1{}%
     \expandafter\hyxmp@commas@to@list@i\expandafter#1#2,,%
163
164 }
```

\hyxmp@commas@to@list@i

Recursively construct macro #1 from comma-separated list #2. Stop if #2 is empty.

```
\next 165 \def\hyxmp@commas@to@list@i#1#2,{%
           \gdef\hyxmp@sublist{#2}%
      166
      167
           \ifx\hyxmp@sublist\@empty
      168
              \let\next=\relax
           \else
      169
              \hyxmp@trimspaces\hyxmp@sublist
      170
      171
              \@cons{#1}{{\hyxmp@sublist}}%
              \def\next{\hyxmp@commas@to@list@i{#1}}%
      172
      173
           \fi
      174
           \next
```

175 }

\xmpcomma Because hyperxmp splits lists at commas, a comma cannot normally be used within a list. We there provide an \mmpcomma macro that can expand to either a true comma or a placeholder character depending on the situation. Here, we bind it to a comma so it can be used in any hyperxmp option, not just those that treat commas specially.

```
176 \def\xmpcomma{,}%
```

\hyxmp@comma

This is what \mmpcomma maps to during list construction. We assume that documents will never otherwise use an ETX (^^C) character in their XMP metadata.

```
177 \bgroup
     \catcode'\^^C=11
   \gdef\hyxmp@comma{^^C}
179
180 \egroup
```

\hyxmp@uscore

This is what _ temporarily maps to during packet construction. Because underscores are replaced by spaces, we need a mechanism to preserve user-specified

underscores (e.g., in email addresses). We assume that documents will never otherwise use an NAK (^^U) character in their XMP metadata.

```
181 \bgroup
182 \catcode'\^^U=11
183 \gdef\hyxmp@uscore{^^U}
184 \egroup
```

\xmpquote

Adobe Acrobat likes to see double quotes around list elements that contain commas when the entire list appears within a single XMP tag (e.g., <pdf:Keywords>). However, it doesn't like to see double quotes around list elements that contain commas when the list is broken up into individual components (i.e., using <rdf:li>tags). We therefore introduce an \xmpquote macro that quotes or doesn't quote its argument based on context. Here, we bind \xmpquote to \relax to prevent it from prematurely quoting or not quoting.

```
185 \let\xmpquote=\relax
```

\mathrmale As a convenience for the user, we define \mathrmale as a category 12 (other) "~" character.

```
186 \bgroup
187 \catcode'\~=12%
188 \gdef\xmptilde{~}%
189 \egroup
```

\XMPTruncateList \hyxmp@temp@str \hyxmp@temp@list

As a workaround for Adobe Acrobat's inability to display author lists correctly (cf. "Acrobat Author bug" on page 6) we introduce a hack that replaces a list with its first element. One can then write "\XMPTruncateList{pdfauthor}" and have Adobe Acrobat display the author list correctly. It's sad that this is necessary, though.

3.3.2 Trimming leading and trailing spaces

To make it easier for XMP processors to manipulate our output we define a \hyxmp@trimspaces macro to strip leading and trailing spaces from various data fields.

\hyxmp@trimspaces

Redefine a macro as its previous value but without leading or trailing spaces. This code—as well as that for its helper macros, \hyxmp@trimb and \hyxmp@trimc—was taken almost verbatim from a solution to an *Around the Bend* puzzle [5]. Inline comments are also taken from the solution text.

199 \catcode'\Q=3

\hyxmp@trimspaces\x redefines \x to have the same replacement text sans leading and trailing space tokens.

200 \newcommand{\hyxmp@trimspaces}[1]{%

Use grouping to emulate a multi-token afterassignment queue.

201 \begingroup

Put "\toks 0 {" into the afterassignment queue.

202 \aftergroup\toks\aftergroup0\aftergroup{%

Apply \hyxmp@trimb to the replacement text of #1, adding a leading \noexpand to prevent brace stripping and to serve another purpose later.

203 \expandafter\hyxmp@trimb\expandafter\noexpand#1Q Q}%

Transfer the trimmed text back into #1.

204 \edef#1{\the\toks0}% 205 }

\hyxmp@trimb

\hyxmp@trimb removes a trailing space if present, then calls \hyxmp@trimc to clean up any leftover bizarre Qs, and trim a leading space. In order for \hyxmp@trimc to work properly we need to put back a Q first.

206 \def\hyxmp@trimb#1 Q{\hyxmp@trimc#1Q}

\hyxmp@trimc

Execute \vfuzz assignment to remove leading space; the \noexpand will now prevent unwanted expansion of a macro or other expandable token at the beginning of the trimmed text. The \endgroup will feed in the \aftergroup tokens after the \vfuzz assignment is completed.

207 \def\hyxmp@trimc#1Q#2{\afterassignment\endgroup \vfuzz\the\vfuzz#1} 208 \catcode'\Q=11

3.3.3 Converting text to XML

The "<", ">", and "&" characters are significant to XML. We therefore need to escape them in any author-supplied text.

\ifhyxmp@unicodetex \hyxmp@unicodetextrue \hyxmp@unicodetexfalse X_TT_EX and LuaT_EX natively support Unicode. We define the conditional \ifhyxmp@unicodetex to check for these so we can properly handle encoding conversions. The trick here is that Unicode T_EX implementations compare decimal 64 to hexadecimal 40 (decimal 64), specified with four carets, and take the TRUE branch; non-Unicode T_EX implementations compare decimal 64 to character "~" (decimal 94), ignore the "~~0040" and the rest of the TRUE branch, and take the FALSE branch.

209 \newif\ifhyxmp@unicodetex 210 \ifnum64='\^^^0040\relax 211 \hyxmp@unicodetextrue 212 \else 213 \hyxmp@unicodetexfalse 214 \fi

This is now a placeholder macro needed only for \@pdfmetalang in the \hyxmp@reencode \begin{document}.

215 \newcommand*{\hyxmp@reencode}[1]{}

\SE->pdfdoc@03 Preserve ETX (^^C), which is normally an invalid character in PDFDocEncoding. We use it in hyperxmp (and specifically in \hyxmp@xmlify below) as a list-element separator.

216 \expandafter\def\csname SE->pdfdoc@03\endcsname{0003}

\SE->pdfdoc@15

Preserve NAK (^^U), which is normally an invalid character in PDFDocEncoding. We use it in hyperxmp (and specifically in \hyxmp@xmlify below) as a placeholder for an underscore character.

217 \expandafter\def\csname SE->pdfdoc@15\endcsname{0015}

\hyxmp@xmlify \hyxmp@xmlified \hyxmp@text

Given a piece of text defined using \pdfstringdef (i.e., with many special characters redefined to have category code 11), set \hyxmp@xmlified to the same text but with all occurrences of "<" replaced with <, all occurrences of ">" replaced with >, and all occurrences of "&" replaced with &.

```
218 \newcommand*{\hyxmp@xmlify}[1]{%
     \gdef\hyxmp@xmlified{}%
 Escaped PDF string \rightarrow PDFDocEncoding/Unicode
     \EdefUnescapeString\hyxmp@text{#1}%
220
     \ifhyxmp@unicodetex
221
 PDFDocEncoding/Unicode \rightarrow UTF-32BE
        \hyxmp@is@unicode\hyxmp@text{%
222
          \StringEncodingConvert
223
          \hyxmp@text\hyxmp@text{utf16be}{utf32be}%
224
225
          \ifxetex
226
            \hyxmp@xetex@crap
227
          \else
228
            \StringEncodingConvert
229
            \hyxmp@text\hyxmp@text{pdfdoc}{utf32be}%
230
231
          \fi
232
       }%
 UTF-32BE \rightarrow UTF-32BE as hex string
        \EdefEscapeHex\hyxmp@text{\hyxmp@text}%
233
 UTF-32BE \rightarrow XML in ASCII
        \edef\hyxmp@text{%
234
          \expandafter
235
       }\expandafter\hyxmp@toxml@unicodetex\hyxmp@text
236
237
       \relax\relax\relax\relax\relax\relax\relax
     \else
238
 PDFDocEncoding/Unicode \rightarrow UTF-8
```

\hyxmp@is@unicode\hyxmp@text{%

239

```
\StringEncodingConvert
240
          \hyxmp@text\hyxmp@text{utf16be}{utf8}%
241
       }{%
242
          \StringEncodingConvert
243
          \hyxmp@text\hyxmp@text{pdfdoc}{utf8}%
244
245
        }%
 UTF-8 \rightarrow UTF-8 as hex string
        \EdefEscapeHex\hyxmp@text{\hyxmp@text}\%
246
 UTF-8 as hex string \rightarrow XML in UTF-8 as hex string
        \edef\hyxmp@text{%
247
248
          \expandafter\hyxmp@toxml\hyxmp@text\@empty\@empty
       }%
249
XML in UTF-8 as hex string \rightarrow XML in UTF-8
        \EdefUnescapeHex\hyxmp@text{\hyxmp@text}%
250
251
     \global\let\hyxmp@xmlified\hyxmp@text
252
253 }
```

\hyxmp@is@unicode \hyxmp@@is@unicode Given a string and two expressions, evaluate the first expression if the string is UTF-16BE-encoded and the second expression if not.

```
254 \begingroup
     \lccode'\<=254 %
255
     \lccode'\>=255 %
256
     \catcode254=12 %
257
258
     \catcode255=12 %
259 \lowercase{\endgroup
260
     \def\hyxmp@is@unicode#1{%
261
        \expandafter\hyxmp@@is@unicode#1<>\@nil
262
     }%
     \def\hyxmp@@is@unicode#1<>#2\@nil{%
263
       \ifx\\#1\\%
264
265
         \expandafter\@firstoftwo
266
          \expandafter\@secondoftwo
267
       \fi
268
     ጉ%
269
270 }
```

\hyxmp@toxml Replace the characters "<", "&", and ">" with XML entities when using a non-native-Unicode TEX (TEX or pdfTEX).

```
271 \def\hyxmp@toxml#1#2{%

272 \ifx#1\@empty

273 \else

274 \ifnum"#1#2='\& %

275 26616D703B% &

276 \else\ifnum"#1#2='\< %

277 266C743B% <

278 \else\ifnum"#1#2='\> %
```

```
279
         2667743B% >
280
```

dvips wraps text when generating most PostScript code but preserves line breaks within strings. Unfortunately, dvips fails to observe the special case in the PostScript specification that "[b]alanced pairs of parentheses in the string require no special treatment" [2]. Consequently, XMP data containing parentheses (e.g., "Copyright (C) 1605 Miguel de Cervantes") confuse dvips into thinking that the string has ended after the closing parenthesis and that line breaks can subsequently be injected safely into the document at arbitrary points for formatting purposes. This leads to erroneous display by PDF viewers, which honor line breaks within XMP tags. The solution is to insert a backslash before all parentheses when in pdfmark-generating mode to convince dvips that the entire XMP packet must be treated as a single, not-to-be-modified string.

```
\@ifundefined{pdfmark}{%
281
            #1#2%
282
283
          }{%
          \ifnum"#1#2='\( %
284
            5C28% \(
285
          \else\ifnum"#1#2='\) %
286
            5C29% \)
287
          \else
288
            #1#2%
289
290
          \fi\fi
291
292
        \fi\fi\fi
293
        \expandafter\hyxmp@toxml
     \fi
294
295 }
```

\hyxmp@text

\hyxmp@toxml@unicodetex Replace the characters "<", "&", and ">" with XML entities when using a native-Unicode TEX (XFTEX or LuaTEX).

```
296 \def\hyxmp@toxml@unicodetex#1#2#3#4#5#6#7#8{%
     \ifx#1\relax
297
     \else
298
       \ifnum"#1#2#3#4#5#6#7#8>127 %
299
         \uccode'\*="#1#2#3#4#5#6#7#8\relax
300
301
         \uppercase{%
302
           \edef\hyxmp@text{\hyxmp@text *}%
         }%
303
       \else\ifnum"#7#8='\< %
304
         \edef\hyxmp@text{\hyxmp@text <}%
305
       \else\ifnum"#7#8='\& %
306
         \edef\hyxmp@text{\hyxmp@text &}%
307
       \else\ifnum"#7#8='\> %
308
         \edef\hyxmp@text{\hyxmp@text >}%
309
       \else\ifnum"#7#8='\ %
310
         \edef\hyxmp@text{\hyxmp@text\space}%
311
312
       \else
```

```
\uccode'\*="#7#8\relax
                    313
                    314
                              \uppercase{%
                                \edef\hyxmp@text{\hyxmp@text *}%
                    315
                    316
                            \fi\fi\fi\fi\fi
                    317
                            \expandafter\hyxmp@toxml@unicodetex
                    318
                    319
                    320 }
  \hyxmp@skipzeros
                   Skip over leading zeroes in the input argument.
                    321 \def\hyxmp@skipzeros#1{%
                    322
                         \ifx#10%
                            \expandafter\hyxmp@skipzeros
                    323
                    324
                         \fi
                    325 }
                 \x In the case of XATEX, the strings defined by \pdfstringdef can contain big
\hyxmp@xetex@crap characters. In this case, the string is treated as Unicode.
        \verb|\hyxmp@try||_{326} \verb|\begingroup||
\hyxmp@crap@result 327 \def\x#1{\endgroup
       \hyxmp@text 328
                         \def\hyxmp@xetex@crap{%
                    329
                            \edef\hyxmp@try{%
                    330
                              \expandafter\hyxmp@SpaceOther\hyxmp@text#1\@nil
                    331
                            }%
                            \let\hyxmp@crap@result=N%
                    332
                            \expandafter\hyxmp@crap@test\hyxmp@try\relax
                    333
                    334
                            \ifx\hyxmp@crap@result Y%
                    335
                              \let\hyxmp@text\@empty
                    336
                              \expandafter\hyxmp@crap@convert\hyxmp@try\relax
                    337
                              \StringEncodingConvert\hyxmp@text\hyxmp@text{pdfdoc}{utf32be}%
                    338
                            \fi
                    339
                         }%
                    340
                    341 }
                    342 \x{ }
\hyxmp@SpaceOther Re-encode all spaces in a string with category code 12 ("other").
                    343 \begingroup
                         \ccite{`}^=12 %
                    344
                         \lccode'\~='\ %
                    345
                    346 \lowercase{\endgroup
                         \def\hyxmp@SpaceOther#1 #2\@nil{%
                    348
                            #1%
                            \int x = 2 \relax
                    349
                              \expandafter\@gobble
                    350
                    351
                            \else
                    352
                              \expandafter\@firstofone
                    353
                    354
```

```
{\hyxmp@SpaceOther#2\@nil}%
                                                                      355
                                                                      356
                                                                                       }%
                                                                      357 }
          \hyxmp@crap@test
                                                                    Determine if we need to treat a string as Unicode.
                                                                      358 \def\hyxmp@crap@test#1{%
                                                                                        \int x#1\relax
                                                                      359
                                                                      360
                                                                                        \else
                                                                                               \ifnum'#1>127 %
                                                                      361
                                                                                                      \let\hyxmp@crap@result=Y%
                                                                      362
                                                                                                      \expandafter\expandafter\expandafter\hyxmp@skiptorelax
                                                                      363
                                                                      364
                                                                                                      \expandafter\expandafter\expandafter\hyxmp@crap@test
                                                                      365
                                                                      366
                                                                                               \fi
                                                                      367
                                                                                        \fi
                                                                      368 }
   \hyxmp@skiptorelax
                                                                      Discard all tokens up to and including the first \relax.
                                                                      369 \def\hyxmp@skiptorelax#1\relax{}
\hyxmp@crap@convert Convert a hexadecimal string to a number.
                              \label{lem:convert} $$ \ \ _{370 \leq hyxmp@crap@convert#1{\%}} $$
                           \hyxmp@text 371
                                                                                        \int x#1\relax
                                                                      372
                                                                                        \else
                                                                      373
                                                                                               \edef\hyxmp@num{\number'#1}%
                                                                      374
                                                                                                \ifnum\hyxmp@num>"FFFFFF %
                                                                                                      \lccode'\!=\intcalcDiv{\hyxmp@num}{\number"1000000}\relax
                                                                      375
                                                                                                      \lowercase{\edef\hyxmp@text{\hyxmp@text!}}%
                                                                      376
                                                                                                      377
                                                                      378
                                                                                                      \edef\hyxmp@text{\hyxmp@text\hyxmp@zero}%
                                                                      379
                                                                      380
                                                                                                \ifnum\hyxmp@num>"FFFF %
                                                                      381
                                                                                                      \label{linear} $$\code'\leq \sum_{\substack{n\in\mathbb{N}\\ number}} \Omega(n) = \sum_{\substack{n\in\mathbb{N}\\ numbe
                                                                      382
                                                                                                      \lowercase{\edef\hyxmp@text{\hyxmp@text!}}%
                                                                      383
                                                                                                      384
                                                                      385
                                                                                                \else
                                                                                                       \edef\hyxmp@text{\hyxmp@text\hyxmp@zero}%
                                                                      386
                                                                      387
                                                                      388
                                                                                                \ifnum\hyxmp@num>"FF %
                                                                                                      \lccode'\!=\intcalcDiv{\hyxmp@num}{\number"100}\relax
                                                                      389
                                                                                                      \lowercase{\edef\hyxmp@text{\hyxmp@text!}}%
                                                                      390
                                                                                                      391
                                                                      392
                                                                                               \else
                                                                                                       \edef\hyxmp@text{\hyxmp@text\hyxmp@zero}%
                                                                      393
                                                                      394
                                                                                                \fi
                                                                                                \ifnum\hyxmp@num>0 %
                                                                      395
                                                                                                      \lccode'\!=\hyxmp@num\relax
                                                                      396
                                                                                                      \lowercase{\edef\hyxmp@text{\hyxmp@text!}}%
                                                                      397
```

```
398
                      \edef\hyxmp@text{\hyxmp@text\hyxmp@zero}%
             399
             400
                     \expandafter\hyxmp@crap@convert
             401
                  \fi
             402
             403 }
\hyxmp@zero
            Define a null character with category code 12 ("other").
             404 \begingroup
                  \catcode0=12 %
                  \gdef\hyxmp@zero{^^00}%
             406
             407 \endgroup
```

UUID generation 3.4

We use a linear congruential generator to produce pseudorandom version 4 UUIDs [8]. True, this method has its flaws but it's simple to implement in TEX and is good enough for producing the XMP xmpMM:DocumentID and xmpMM:InstanceID fields.

\hyxmp@modulo@a

Replace the contents of \Otemponta with the contents modulo #1. Note that \@tempcntb is overwritten in the process.

```
408 \def\hyxmp@modulo@a#1{%
     \@tempcntb=\@tempcnta
     \divide\@tempcntb by #1
410
     \multiply\@tempcntb by #1
411
     \advance\@tempcnta by -\@tempcntb
412
413 }
```

\hyxmp@big@prime Define a couple of large prime numbers that can still be stored in a TFX counter.

```
415 \def\hyxmp@big@prime@ii{536870027}
```

\hyxmp@seed@rng Seed hyperxmp's random-number generator from a given piece of text.

```
\verb|\hyxmp@one@token||_{416} $$ \def\hyxmp@seed@rng#1{%}
                         \@tempcnta=\hyxmp@big@prime
                   417
                         \futurelet\hyxmp@one@token\hyxmp@seed@rng@i#1\@empty
                   418
                   419 }
```

\hyxmp@one@token

 $\mbox{hyxmp@seed@rng@i}$ Do all of the work for $\mbox{hyxmp@seed@rng}$. For each character code c of the input text, assign $\ensuremath{\texttt{Qtempcnta}} + c \pmod{\texttt{hyxmp@big@prime}}$.

```
\next 420 \def\hyxmp@seed@rng@i{%
           \ifx\hyxmp@one@token\@empty
      421
              \let\next=\relax
      422
      423
           \else
      424
              \def\next##1{%
      425
                \multiply\@tempcnta by 3
                \advance\@tempcnta by '##1
      426
                \hyxmp@modulo@a{\hyxmp@big@prime}%
      427
```

```
\futurelet\hyxmp@one@token\hyxmp@seed@rng@i
428
        }%
429
     \fi
430
     \next
431
432 }
```

\hyxmp@set@rand@num \hyxmp@rand@num

Advance \hyxmp@rand@num to the next pseudorandom number in the sequence. Specifically, we assign $\mbox{hyxmp@rand@num} \leftarrow 3 \cdot \mbox{hyxmp@rand@num} +$ \hyxmp@big@prime@ii (mod \hyxmp@big@prime). Note that both \@tempcnta and \@tempcntb are overwritten in the process.

```
433 \def\hyxmp@set@rand@num{%
434
     \@tempcnta=\hyxmp@rand@num
435
     \multiply\@tempcnta by 3
     \advance\@tempcnta by \hyxmp@big@prime@ii
436
     \hyxmp@modulo@a{\hyxmp@big@prime}%
437
     \xdef\hyxmp@rand@num{\the\@tempcnta}%
438
439 }
```

\hyxmp@append@hex

Append a randomly selected hexadecimal digit to macro #1. Note that both \Otempcnta and \Otempcntb are overwritten in the process.

```
440 \def\hyxmp@append@hex#1{%
     \hyxmp@set@rand@num
441
     \@tempcnta=\hyxmp@rand@num
442
443
     \hyxmp@modulo@a{16}%
444
     \ifnum\@tempcnta<10
       \xdef#1{#1\the\@tempcnta}%
445
     \else
446
```

There must be a better way to handle the numbers 10-15 than with \ifcase.

```
447
         \advance\@tempcnta by -10
        \ifcase\@tempcnta
448
           \xdef#1{#1a}%
449
           \c \ \or\xdef#1{#1b}%
450
           \c \ \or\xdef#1{#1c}%
451
452
           \c\or\xdef#1{#1d}%
453
           \c \fi = 1{\#1e}%
           \c \ \or\xdef#1{#1f}%
454
455
        \fi
      \fi
456
457 }
```

\hyxmp@append@hex@iii Invoke \hyxmp@append@hex three times.

```
458 \def\hyxmp@append@hex@iii#1{%
     \hyxmp@append@hex#1%
     \hyxmp@append@hex#1%
460
461
     \hyxmp@append@hex#1%
462 }
```

\hyxmp@append@hex@iv Invoke \hyxmp@append@hex four times.

```
463 \def\hyxmp@append@hex@iv#1{%
464 \hyxmp@append@hex@iii#1%
465 \hyxmp@append@hex#1%
466 }
```

\hyxmp@create@uuid

```
467 \def\hyxmp@create@uuid#1{%
     \def#1{uuid:}%
468
     \hyxmp@append@hex@iv#1%
469
     \hyxmp@append@hex@iv#1%
470
     \g@addto@macro#1{-}%
471
     \hyxmp@append@hex@iv#1%
472
     \g@addto@macro#1{-4}%
473
     \hyxmp@append@hex@iii#1%
474
     \g@addto@macro#1{-}%
 Randomly select one of "8", "9", "a", or "b".
     \hyxmp@set@rand@num
476
477
     \@tempcnta=\hyxmp@rand@num
     \hyxmp@modulo@a{4}%
478
479
     \ifcase\@tempcnta
480
        \g@addto@macro#1{8}%
        \or\g@addto@macro#1{9}%
481
        \or\g@addto@macro#1{a}%
482
483
        \or\g@addto@macro#1{b}%
     \fi
484
     \hyxmp@append@hex@iii#1%
485
486
     \g@addto@macro#1{-}%
     \hyxmp@append@hex@iv#1%
487
     \hyxmp@append@hex@iv#1%
     \hyxmp@append@hex@iv#1%
489
490 }
```

\hyxmp@def@DocumentID \hyxmp@DocumentID Seed the random-number generator with a function of the current filename, PDF document title, and PDF author, then invoke \hyxmp@create@uuid to define \hyxmp@DocumentID as a random UUID.

```
491 \newcommand*{\hyxmp@def@DocumentID}{%
492 \edef\hyxmp@seed@string{\jobname:\@pdftitle:\@pdfauthor}%
493 \expandafter\hyxmp@seed@rng\expandafter{\hyxmp@seed@string}%
494 \edef\hyxmp@rand@num{\the\@tempcnta}%
495 \hyxmp@create@uuid\hyxmp@DocumentID
496 }
```

\hyxmp@def@InstanceID \hyxmp@InstanceID

Seed the random-number generator with a function of the current filename, PDF document title, PDF author, and the current timestamp, then invoke \hyxmp@create@uuid to define \hyxmp@InstanceID as a random UUID.

```
497 \newcommand*{\hyxmp@def@InstanceID}{%
498 \edef\hyxmp@seed@string{\jobname:\@pdftitle:\@pdfauthor:\hyxmp@today}%
499 \expandafter\hyxmp@seed@rng\expandafter{\hyxmp@seed@string}%
500 \edef\hyxmp@rand@num{\the\@tempcnta}%
501 \hyxmp@create@uuid\hyxmp@InstanceID
502 }
```

3.5 Constructing the XMP packet

An XMP packet "shall consist of the following, in order: a header PI, the serialized XMP data model (the XMP packet) with optional white-space padding, and a trailer PI" [4]. ("PI" is an abbreviation for "processing instructions"). The serialized XMP includes blocks of XML for various XMP schemata: Adobe PDF (Section 3.5.2), Dublin Core (Section 3.5.3), XMP Rights Management (Section 3.5.4), XMP Media Management (Section 3.5.5), XMP Basic (Section 3.5.6), Photoshop (Section 3.5.7), IPTC Photo Metadata (Section 3.5.8), and PDF/A Identification (Section 3.5.9). The \hyxmp@construct@packet macro (Section 3.5.10) constructs the XMP packet into \hyxmp@xml. It first writes the appropriate XML header, then calls the various schema-writing macros, then injects \hyxmp@padding as padding, and finally writes the appropriate XML trailer.

3.5.1 XMP utility functions

\hyxmp@add@to@xml

Given a piece of text, replace all underscores with category-code 11 ("other") spaces and all ^C characters with commas, then append the result to the \hyxmp@xml macro.

```
503 \newcommand*{\hyxmp@add@to@xml}[1]{%
                  \bgroup
             504
             505
                     \@tempcnta=0
             506
                     \loop
             507
                       \lccode\@tempcnta=\@tempcnta
             508
                       \advance\@tempcnta by 1
                       \ifnum\@tempcnta<256
             509
                     \repeat
             510
                     \lccode'\_='\ \relax
             511
                     \lccode'\^^C='\,\relax
             512
                     \lccode'\^^U='\_\relax
             513
                     \lowercase{\xdef\hyxmp@new@xml{#1}}%
             514
                     \xdef\hyxmp@xml{\hyxmp@xml\hyxmp@new@xml}%
             515
             516
                  \egroup
             517 }
\hyxmp@hash Define a category-code 11 ("other") version of the "#" character.
             518 \bgroup
             519 \catcode '\#=11
             520 \gdef\hyxmp@hash{#}
             521 \egroup
```

\hyxmp@padding \hyxmp@xml The XMP specification recommends leaving approximately 2000 bytes of whitespace at the end of each XMP packet to facilitate editing the packet in place [4]. \hyxmp@padding is defined to contain 32 lines of 50 spaces and a newline apiece for a total of 1632 characters of whitespace.

```
522 \bgroup
                                 \xdef\hyxmp@xml{}%
523
                                 \hyxmp@add@to@xml{%
524
525 _
                              }
526
                                \xdef\hyxmp@padding{\hyxmp@xml}%
527
528 \egroup
529 \xdef\hyxmp@padding{\hyxmp@padding\hyxmp@padding}
530 \xdef\hyxmp@padding{\hyxmp@padding\hyxmp@padding}
531 \xdef\hyxmp@padding{\hyxmp@padding\hyxmp@padding}
532 \ensuremath{\mbox{\mbox{$1$}}} 132 \ensuremath{\mbox{\mbox{$2$}}} 132 \ensuremath{\mbox{\mbox{$2$}}} 132 \ensuremath{\mbox{\mbox{$2$}}} 132 \ensuremath{\mbox{$2$}} 
533 \xdef\hyxmp@padding{\hyxmp@padding\hyxmp@padding}
```

\hyxmp@pdf@to@xmp@date

Convert a timestamp from PDF's D:YYYYMMDDhhmmss-TT'tt' format (e.g., D:20140924231019-06'00') to XMP's YYYY-MM-DDThh:mm:ss+TT:tt format (e.g., 2014-09-24T23:10:19-06:00) [4]. This macro is fully expandable.

```
534 \def\hyxmp@pdf@to@xmp@date#1:#2#3#4#5#6#7#8#9{%
535 #2#3#4#5-#6#7-#8#9%
536 \hyxmp@parse@time
537 }
```

\hyxmp@parse@time

This is a helper function for \hyxmp@pdf@to@xmp@date.\hyxmp@pdf@to@xmp@date proper parses only the year, month, and day then calls \hyxmp@parse@time. \hyxmp@parse@time parses the hours, minutes, and seconds then calls \hyxmp@parse@tz@char.

```
538 \def\hyxmp@parse@time#1#2#3#4#5#6{%
539    T#1#2:#3#4:#5#6%
540    \hyxmp@parse@tz@char
541 }
```

\hyxmp@parse@tz@char

This is another helper function for $\mbox{hyxmp@pdf@to@xmp@date}$. So far, the date and time have been parsed. $\mbox{hyxmp@parse@tz@char}$ parses the first character of the timezone descriptor. This can be one of "+" for eastern timezones (UTC+x, including Asia, Oceania, and most of Europe), "-" for western timezones (UTC-x, primarily the Americas), or "Z" for Zulu time (UTC+0). Timezones beginning with "+" or "-" are followed by an offset in hours and minutes (parsed by $\mbox{hyxmp@parse@tz}$; timezones beginning with "Z" are not.

```
542 \def\hyxmp@parse@tz@char#1{%

543 #1%

544 \ifx#1-%

545 \expandafter\hyxmp@parse@tz

546 \else

547 \ifx#1+%

548 \expandafter\hyxmp@parse@tz
```

```
\fi
549
      \fi
550
551 }
```

\hyxmp@parse@tz

This is the final helper function for \hyxmp@pdf@to@xmp@date. It parses the piece of the timezone comprising the offset from Coordinated Universal Time, measured in hours and minutes.

```
552 \def\hyxmp@parse@tz#1'#2'{%}
     #1:#2%
554 }
```

\hyxmp@today@define

Use TrX's \year, \month, and \day primitives to define \hyxmp@today as today's date in YYYY-MM-DD format.

```
555 \def\hyxmp@today@define{%
     \xdef\hyxmp@today{\the\year}%
556
557
     \ifnum\month<10
558
        \xdef\hyxmp@today{\hyxmp@today-0\the\month}%
559
     \else
        \xdef\hyxmp@today{\hyxmp@today-\the\month}%
560
     \fi
561
     \int \frac{10}{10}
562
       \xdef\hyxmp@today{\hyxmp@today-0\the\day}%
563
564
       \\d \hyxmp@today{\hyxmp@today-\the\day}%
565
566
     \fi
567 }
```

\hyxmp@today

Define \hyxmp@today as the current date and (if available) time and timezone in XMP Date format [4].

```
568 \expandafter\ifx\csname pdfcreationdate\endcsname\relax
```

\hyxmp@today@define 569

570 \else

\edef\hyxmp@today{\expandafter\hyxmp@pdf@to@xmp@date\pdfcreationdate}% 571 572 \fi

\hyxmp@x@default Define an x-default string that we can use in comparisons with \@pdfmetalang. 573 \newcommand*{\hyxmp@x@default}{x-default}

The Adobe PDF schema

\hyxmp@pdf@schema

Add properties defined by the Adobe PDF schema to the \hyxmp@xml macro.

574 \newcommand*{\hyxmp@pdf@schema}{%

Add a block of XML to \hyxmp@xml that lists the document's keywords (the pdf:Keywords property), the tools used to produce the PDF file (the pdf:Producer property), and the version of the PDF standard adhered to (the pdf:PDFVersion property). Unlike most of the other schemata that hyperxmp supports, the Adobe PDF schema is always included in the document, even if all of its keys are empty. This is because PDF/A-1b requires the keywords and producer to be the same in

the XMP metadata and the PDF metadata. Because hyperref always specifies the Keywords and Producer fields, even when they're empty, hyperxmp has to follow suit and define pdf:Keywords and pdf:Producer in the XMP packet.

```
\hyxmp@add@to@xml{%
576 _____<rdf:Description rdf:about=""^^J%
577 _____xmlns:pdf="http://ns.adobe.com/pdf/1.3/">^^J%
     \hyxmp@add@simple@var{pdf:Keywords}{@pdfkeywords}%
579
580
     \hyxmp@add@simple@var{pdf:Producer}{@pdfproducer}%
     \@ifundefined{pdfminorversion}{}{%
581
       \hyxmp@add@simple{pdf:PDFVersion}{1.\the\pdfminorversion}%
582
     }%
583
     \hyxmp@add@to@xml{%
584
585 ____</rdf:Description>^^J%
586
    }%
587 }
```

\hyxmp@string

\hyxmp@add@simple Given an XMP tag (#1) and a string (#2), if the string is nonempty, add a begin tag, the string, and an end tag to the packet. The "simple" in the macro name indicates that the string is output without variations for different languages.

```
588 \newcommand*{\hyxmp@add@simple}[2]{%
     \edef\hyxmp@string{#2}%
589
590
     \ifx\hyxmp@string\@empty
591
592
       \hyxmp@xmlify{\hyxmp@string}%
593
       \hyxmp@add@to@xml{%
    _____<#1>\hyxmp@xmlified</#1>^^J%
594
       }%
595
596
     \fi
597 }
```

\hyxmp@add@simple@var

Given an XMP tag (#1) and a variable name (#2), if the string is defined, add a begin tag, the string, and an end tag to the packet. The "simple" in the macro name indicates that the string is output without variations for different languages. \hyxmp@add@simple@var differs from \hyxmp@add@simple in that the former includes defined but empty values in the XMP packet while the latter excludes both undefined and defined but empty values.

```
598 \newcommand*{\hyxmp@add@simple@var}[2]{%
599
     \expandafter\ifx\csname#2\endcsname\relax
600
     \else
       \hyxmp@xmlify{\csname#2\endcsname}%
601
       \hyxmp@add@to@xml{%
602
   _____<#1>\hyxmp@xmlified</#1>^^J%
603
604
       }%
605
     \fi
606 }
```

3.5.3 The Dublin Core schema

\hyxmp@rdf@dc Given a Dublin Core property (#1) and a macro containing some \pdfstringdef-defined text (#2), append the appropriate block of XML to the \hyxmp@xml macro but only if #2 is non-empty.

```
607 \newcommand*{\hyxmp@rdf@dc}[2]{%
                                  \frak{1}{2}\end{cempty}
608
609
                                  \else
                                                 \hyxmp@xmlify{#2}%
610
                                                \hyxmp@add@to@xml{%
612 _____<dc:#1>^^J%
613 _____<rdf:Alt>^^J%
614
                                               }%
615
                                                \ifx\@pdfmetalang\hyxmp@x@default
616
617
                                                             \hyxmp@add@to@xml{%
618 \ \_\_\_ < rdf: li \ xml: lang="\qpdfmetalang">\hyxmp@xmlified</rdf: li>^^J\% = li>^hyxmp@xmlified</rdf: li>^hyxmp@xmli
619
                                                  }%
                                                 \fi
620
                                                \hyxmp@add@to@xml{%
621
622 \ \_\_\_ < rdf: li \ xml: lang="\hyxmp@x@default">\hyxmp@xmlified</rdf: li>^^J\% = li>^hyxmp@xmlified</rdf: li>^hyxmp@xmlified</rd>
623 _____</rdf:Alt>^^J%
624 _____</dc:#1>^^J%
625
626 \fi%
627 }%
```

\hyxmp@list@to@xml

Given a Dublin Core property (#1), an RDF array (#2), and a macro containing a comma-separated list (#3), append the appropriate block of XML to the \hyxmp@xml macro but only if #3 is non-empty.

```
628 \newcommand*{\hyxmp@list@to@xml}[3]{%
629 \ifx#3\@empty
630 \else
631 \hyxmp@add@to@xml{%
632 _____<dc:#1>^^J%
633 ____<rdf:#2>^^J%
634 }%
635 \bgroup
```

\@elt Re-encode the text from Unicode if necessary. Then redefine \@elt to XML-ify each element of the list and append it to \hyxmp@xmlified.

```
\hyxmp@xmlify{#3}%
636
         \hyxmp@commas@to@list\hyxmp@list{\hyxmp@xmlified}%
637
         \def\@elt##1{%}
638
           \hyxmp@add@to@xml{%
639
          _____<rdf:li>##1</rdf:li>^^J%
640 _
           }%
641
642
         }%
         \hyxmp@list
643
```

```
644 \egroup
645 \hyxmp@add@to@xml{%
646 _____</rdf:#2>^^J%
647 _____</dc:#1>^^J%
648 }%
649 \fi
650}
```

\hyxmp@dc@schema

Add properties defined by the Dublin Core schema to the \hyxmp@xml macro. Specifically, we add entries for the dc:title property if the author specified a pdftitle, the dc:description property if the author specified a pdfsubject, the dc:rights property if the author specified a pdfsubject, the dc:rights property if the author specified a pdfauthor, the dc:subject property if the author specified pdfkeywords, and the dc:language property if the author specified pdflang. We also specify the dc:date property using the date the document was run through LATEX and the dc:source property using the base name of the source file with .tex appended.

```
651 \newcommand*{\hyxmp@dc@schema}{%
    \hyxmp@add@to@xml{%
653 _____<rdf:Description rdf:about=""^^J%
654 _____xmlns:dc="http://purl.org/dc/elements/1.1/">^^J%
655 _____<dc:format>application/pdf</dc:format>^^J%
    }%
656
657
     \hyxmp@rdf@dc{title}{\@pdftitle}%
     \hyxmp@rdf@dc{description}{\@pdfsubject}%
658
659
     \hyxmp@rdf@dc{rights}{\@pdfcopyright}%
660
     \hyxmp@list@to@xml{creator}{Seq}{\hyxmp@pdfauthor}%
     \hyxmp@list@to@xml{subject}{Bag}{\hyxmp@pdfkeywords}%
662
     \hyxmp@list@to@xml{date}{Seq}{\hyxmp@today}%
663
     \hyxmp@list@to@xml{language}{Bag}{\@pdflang}%
     \hyxmp@add@simple{dc:source}{\jobname.tex}%
664
665
     \hyxmp@add@to@xml{%
666 _____</rdf:Description>^^J%
667
    }%
668 }
```

3.5.4 The XMP Rights Management schema

\hyxmp@xmpRights@schema

Add properties defined by the XMP Rights Management schema to the \hyxmp@xml macro. Currently, these are only the xmpRights:Marked property and the xmpRights:WebStatement property. If the author specified a copyright statement we mark the document as copyrighted. If the author specified a license statement we include the URL in the metadata.

669 \newcommand*{\hyxmp@xmpRights@schema}{%

\hyxmp@legal Set \hyxmp@rights to YES if either pdfcopyright or pdflicenseurl was specified.

```
670 \let\hyxmp@rights=\@empty
671 \ifx\@pdflicenseurl\@empty
```

```
\else
672
673
       \def\hyxmp@rights{YES}%
     \fi
674
     \ifx\@pdfcopyright\@empty
675
676
677
        \def\hyxmp@rights{YES}%
678
Include the license-statement URL and/or the copyright indication. The copyright
 statement itself is included by \hyxmp@dc@schema in Section 3.5.3.
     \ifx\hyxmp@rights\@empty
679
680
     \else
 Header
        \hyxmp@add@to@xml{%
681
682 \, \mbox{\_----} \mbox{<rdf:Description rdf:about=""^^J\% }
683 _____xmlns:xmpRights="http://ns.adobe.com/xap/1.0/rights/">^^J%
 Copyright indication
685
       \ifx\@pdfcopyright\@empty
686
          \hyxmp@add@to@xml{%
687
         ____<xmpRights:Marked>True</xmpRights:Marked>^^J%
688
         }%
689
690
       \fi
License URL
        \hyxmp@add@simple{xmpRights:WebStatement}{\@pdflicenseurl}%
691
Trailer
       \hyxmp@add@to@xml{%
692
693 _____</rdf:Description>^^J%
694
       }%
695
     \fi
696 }
```

3.5.5 The XMP Media Management schema

\hyxmp@mm@schema

Add properties defined by the XMP Media Management schema to the \hyxmp@xml macro. According to the XMP specification, the xmpMM:DocumentID property is supposed to uniquely identify a document, and the xmpMM:InstanceID property is supposed to change with each save operation [4]. As seen in Section 3.4, we do what we can to honor this intention from within a TFX-based workflow.

```
697 \gdef\hyxmp@mm@schema{%
698 \hyxmp@def@DocumentID
699 \hyxmp@def@InstanceID
700 \hyxmp@add@to@xml{%
701 _____<rdf:Description rdf:about=""^^J%
702 _____xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/">^^J%
703 _____<xmpMM:DocumentID>\hyxmp@DocumentID</xmpMM:DocumentID>^^J%
```

```
704 _____<xmpMM:InstanceID>\hyxmp@InstanceID</xmpMM:InstanceID>^^J%
705 ____</rdf:Description>^^J%
706 }%
707 }
```

3.5.6 The XMP Basic schema

\hyxmp@xmp@basic@schema

Add properties defined by the XMP Basic schema to the \hyxmp@xml macro. These include a bunch of dates (all set to the same value) and the base URL for the document if specified with baseurl.

```
708 \newcommand*{\hyxmp@xmp@basic@schema}{%
     \hyxmp@add@to@xml{%
710 _____<rdf:Description rdf:about=""^^J%
711 _____xmlns:xmp="http://ns.adobe.com/xap/1.0/">^^J%
712
     \hyxmp@add@simple{xmp:CreateDate}{\hyxmp@today}%
713
     \hyxmp@add@simple{xmp:ModifyDate}{\hyxmp@today}%
714
     \hyxmp@add@simple{xmp:MetadataDate}{\hyxmp@today}%
715
     \hyxmp@add@simple{xmp:CreatorTool}{\@pdfcreator}%
716
     \hyxmp@add@simple{xmp:BaseURL}{\@baseurl}%
717
     \hyxmp@add@to@xml{%
718
719 _____</rdf:Description>^^J%
   }%
720
721 }
```

3.5.7 The Photoshop schema

\hyxmp@photoshop@schema \hyxmp@photoshop@data Add properties defined by the Photoshop schema to the \hyxmp@xml macro. We currently support only the photoshop:AuthorsPosition and photoshop:CaptionWriter properties.

```
722 \gdef\hyxmp@photoshop@schema{%
     \edef\hyxmp@photoshop@data{\@pdfauthortitle\@pdfcaptionwriter}%
     \ifx\hyxmp@photoshop@data\@empty
724
     \else
725
       \hyxmp@add@to@xml{%
726
    ____<rdf:Description rdf:about=""^^J%
727
728 _
    _____xmlns:photoshop="http://ns.adobe.com/photoshop/1.0/">^^J%
729
       }%
730
731
     \hyxmp@add@simple{photoshop:AuthorsPosition}{\@pdfauthortitle}%
     \hyxmp@add@simple{photoshop:CaptionWriter}{\@pdfcaptionwriter}%
732
     \ifx\hyxmp@photoshop@data\@empty
733
734
735
       \hyxmp@add@to@xml{%
736 _____</rdf:Description>^^J%
737
       }%
738
    \fi
739 }
```

3.5.8 The IPTC Photo Metadata schema

Lines in multiline fields are separated by \mmplinesep in the generated XML. This defaults to an LF (^^J) character but written as an XML character entity for consistency across operating systems.

```
740 \begingroup
      \colored{catcode'} = 12
741
      \colored{catcode'} = 12
742
      \gdef\xmplinesep{
}
744 \endgroup
```

\hyxmp@list@to@lines Given a property (#1) and a macro containing a comma-separated list (#2), replace commas with \xmplinesep. Do nothing it the list is empty.

```
745 \newcommand*{\hyxmp@list@to@lines}[2]{%
     \ifx#2\@empty
746
     \else
747
748
       \bgroup
         \hyxmp@add@to@xml{%
750 ____<#1>%
751
         }%
```

\OeltOfirst The first element of the list is output as is.

```
752
         \def\@elt@first##1{%
753
            \hvxmp@add@to@xml{##1}%
            \let\@elt=\@elt@rest
754
         }%
755
```

\@elt@rest The remaining elements of the list are output with a preceding line separator (\xmplinesep).

```
756
         \def\@elt@rest##1{%
           \hyxmp@add@to@xml{\xmplinesep##1}%
757
758
```

\@elt Re-encode the text from Unicode if necessary. Then redefine \@elt to insert a line separator between terms.

```
\let\@elt=\@elt@first
759
        \hyxmp@xmlify{#2}%
760
761
        \hyxmp@commas@to@list\hyxmp@list{\hyxmp@xmlified}%
        \hyxmp@list
762
        763
764
      \egroup
765
    \fi
766 }
```

\hyxmp@photometa@data

\hyxmp@photometa@schema Add properties defined by the IPTC Photo Metadata schema [6] to the \hyxmp@xml macro. We currently support only the contact-information lptc4xmpCore:CreatorContactInfo/CiAdrExtadr, details structure, viz. $_{
m the}$ lptc4xmpCore:CreatorContactInfo/CiAdrCity, Iptc4xmpCore:CreatorContactInfo/ CiAdrRegion, Iptc4xmpCore:CreatorContactInfo/CiAdrPcode, $\label{linear_contact} Iptc4xmpCore: Creator Contact Info/CiAdr Ctry, & Iptc4xmpCore: Creator Contact Info/CiTelWork, & Iptc4xmpCore: Creator Contact Info/CiEmailWork, & and Iptc4xmpCore: Creator Contact Info/CiUrlWork properties. \\$

```
767 \gdef\hyxmp@photometa@schema{%
     \edef\hyxmp@photometa@data{%
       \@pdfcontactaddress
769
770
       \@pdfcontactcity
       \@pdfcontactregion
771
772
       \@pdfcontactpostcode
773
       \@pdfcontactcountry
774
       \@pdfcontactphone
775
       \@pdfcontactemail
776
       \@pdfcontacturl
777
     \ifx\hyxmp@photometa@data\@empty
778
779
       \hyxmp@iptc@extensions
780
       \hyxmp@add@to@xml{%
781
782 _____<rdf:Description rdf:about=""^^J\%
783 _____xmlns:Iptc4xmpCore="http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/"^J%
784 _____xmlns:IptcContInfo="http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactinfo/">^^J%
785 _____<Iptc4xmpCore:CreatorContactInfo rdf:parseType="Resource">^^J%
786
       }%
787
     \fi
     \hyxmp@list@to@lines{IptcContInfo:CiAdrExtadr}{\@pdfcontactaddress}%
788
     \hyxmp@add@simple{IptcContInfo:CiAdrCity}{\@pdfcontactcity}%
789
     \hyxmp@add@simple{IptcContInfo:CiAdrRegion}{\@pdfcontactregion}%
790
791
     \hyxmp@add@simple{IptcContInfo:CiAdrPcode}{\@pdfcontactpostcode}%
     \hyxmp@add@simple{IptcContInfo:CiAdrCtry}{\@pdfcontactcountry}%
```

\xmplinesep

The IPTC standard states that sets of telephone numbers, email addresses, and URLs for the contact person or institution, "[m]ay have to be separated by a comma in the user interface" [6]. This is rather ambiguous: Does the comma appear *only* in the user interface or also in the generated XML? Here we assume the latter interpretation and temporarily redefine \xmplinesep as a comma and use \hyxmp@list@to@lines to insert the data. Unlike \hyxmp@add@simple, this approach trims all spaces surrounding commas.

```
\bgroup
793
       \def\xmplinesep{,}%
794
       \hyxmp@list@to@lines{IptcContInfo:CiTelWork}{\@pdfcontactphone}%
795
       \hyxmp@list@to@lines{IptcContInfo:CiEmailWork}{\@pdfcontactemail}%
796
       \hyxmp@list@to@lines{IptcContInfo:CiUrlWork}{\@pdfcontacturl}%
797
798
     \ifx\hyxmp@photometa@data\@empty
799
800
       \hyxmp@add@to@xml{%
801
802 _____</Iptc4xmpCore:CreatorContactInfo>^^J%
803 ____</rdf:Description>^^J%
804
       }%
```

```
805 \fi
806 }
```

\hyxmp@iptc@extensions

Because IPTC metadata are not recognized by the PDF/A standard, PDF/A conversion would normally fail for documents that utilize \pdfcontactaddress, \pdfcontactcity, etc. However, there exists a technique, described in a PDF Association technical note [10], for describing nonstandard XMP metadata within the XMP packet itself. We use that technique here to describe all of the metadata that \hyxmp@photometa@schema can produce. Doing so enables the document to be converted to PDF/A format.

```
807 \newcommand*{\hyxmp@iptc@extensions}{%
            \hyxmp@add@to@xml{%
809 _____<rdf:Description rdf:about=""^^J%
810 _____xmlns:pdfaExtension="http://www.aiim.org/pdfa/ns/extension/"^^J%
811 _____xmlns:pdfaSchema="http://www.aiim.org/pdfa/ns/schema\hyxmp@hash"^^J%
812 _____xmlns:pdfaProperty="http://www.aiim.org/pdfa/ns/property\hyxmp@hash"^J%
813 _____xmlns:pdfaType="http://www.aiim.org/pdfa/ns/type\hyxmp@hash"^^J%
814 _____xmlns:pdfaField="http://www.aiim.org/pdfa/ns/field\hyxmp@hash">^^J%
815 _____<pdfaExtension:schemas>^^J\%
816 _____<rdf:Bag>^^J%
817 _____<rdf:li rdf:parseType="Resource">^^J%
819 \ \_\_\_\_ < pdfaSchema: namespaceURI > http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/ < /pdfaSchema: namespaceURI > http:/
820~\_\_\_<pdfaSchema:prefix>Iptc4xmpCore</pdfaSchema:prefix>^^J\%
821 _____<pdfaSchema:property>^^J%
822 _____<rdf:Seq>^^J%
823 _____<rdf:li rdf:parseType="Resource">^^J%
824 ______<pdfaProperty:name>CreatorContactInfo</pdfaProperty:name>^^J%
825 ______<pdfaProperty:valueType>contactinfo</pdfaProperty:valueType>^^J%
826 ______<pdfaProperty:category>external</pdfaProperty:category>^^J%
827 _____<pdfaProperty:description>contact information for the document's creator</p
828 _____</rdf:li>^^J%
829 _____</rdf:Seq>^^J%
830 _____</pdfaSchema:property>^^J%
831 _____<pdfaSchema:valueType>^^J%
832 _____<rdf:Seq>^^J%
833 _____<rdf:li rdf:parseType="Resource">^^J%
835 \ \_\_\_\_<pdfaType:namespaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactingspaceURI>http://iptc.org/std/Iptc4xm
836 ______<pdfaType:prefix>IptcContInfo</pdfaType:prefix>^^J%
837 ______<pdfaType:description>contact information</pdfaType:description>^^J%
838 _____<pdfaType:field>^^J%
839 _____<rdf:Seq>^^J%
           }%
840
             \hyxmp@text@resource{CiAdrExtadr}{contact address}%
841
             \hyxmp@text@resource{CiAdrCity}{contact city}%
842
             \hyxmp@text@resource{CiAdrRegion}{contact region}%
843
844
             \hyxmp@text@resource{CiAdrPcode}{contact postal code}%
             \hyxmp@text@resource{CiAdrCtry}{contact country}%
```

```
\hyxmp@text@resource{CiTelWork}{contact telephone number}%
                  846
                      \hyxmp@text@resource{CiEmailWork}{contact email address}%
                  847
                      \hyxmp@text@resource{CiUrlWork}{contact url}%
                  848
                      \hyxmp@add@to@xml{%
                  849
                  850 _____</rdf:Seq>^^J%
                  851 _____</pdfaType:field>^^J%
                  852 _____</rdf:li>^^J%
                  853 _____</rdf:Seq>^^J%
                  854 _____</pdfaSchema:valueType>^^J%
                  855 _____</rdf:li>^^J%
                  856 _____</rdf:Bag>^^J%
                  857 _____</pdfaExtension:schemas>^^J%
                  858 _____</rdf:Description>^^J%
                  859 }%
                  860 }
\hyxmp@text@resource
                  Output a single Text resource given its name and description.
                  861 \newcommand*{\hyxmp@text@resource}[2]{%
                  862 \hyxmp@add@to@xml{%
                  863 _____<rdf:li rdf:parseType="Resource">^^J%
                  864 _____<pdfaField:name>#1</pdfaField:name>^^J%
                  865 _____<pdfaField:valueType>Text</pdfaField:valueType>^^J%
                  866 \ \_\_\_\_<pdfaField:description>#2</pdfaField:description>^^J\%
                  867 _____</rdf:li>^^J%
                  868
                  869 }
```

3.5.9 The PDF/A Identification schema

\hyxmp@pdfa@id@schema

Add properties defined by the PDF/A Identification schema [9] to the \hyxmp@xml macro. These properties identify a document as conforming to a particular PDF/A standard. Currently, we assume PDF/A-1b if any PDF/A compliance is detected.

```
870 \newcommand*{\hyxmp@pdfa@id@schema}{%
     \ifHy@pdfa
871
       \hyxmp@add@to@xml{%
872
873 _____<rdf:Description rdf:about=""^^J%
874 _____xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/">^^J%
875
       \hyxmp@add@simple{pdfaid:part}{1}%
876
       \hyxmp@add@simple{pdfaid:conformance}{B}%
877
       \hyxmp@add@to@xml{%
878
879 _____</rdf:Description>^^J%
       }%
    \fi
881
882 }
```

3.5.10 Combining schemata into an XMP packet

\hyxmp@bom Define a macro for the Unicode byte-order marker (BOM).

```
883 \begingroup
     \ifhyxmp@unicodetex
884
        \lccode'\!="FEFF %
885
        \lowercase{%
886
          \gdef\hyxmp@bom{!}
887
888
       }%
889
     \else
        \catcode'\^^ef=12
890
        \catcode'\^^bb=12
891
        \catcode'\^^bf=12
892
        \gdef\hyxmp@bom{^^ef^^bb^^bf}%
893
     \fi
894
895 \endgroup
```

\hyxmp@construct@packet \hyxmp@xml

Successively add XML data to \hyxmp@xml until we have something we can insert into the document's PDF catalog.

```
896 \def\hyxmp@construct@packet{%
897
     \gdef\hyxmp@xml{}%
     \hyxmp@add@to@xml{<?xpacket begin="\hyxmp@bom" %
899 id="W5MOMpCehiHzreSzNTczkc9d"?>^^J%
900 <x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="3.1-702">^^J%
901 ___<rdf:RDF
902 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns\hyxmp@hash">^^J%
     }%
903
904
     \hyxmp@pdf@schema
     \hyxmp@xmpRights@schema
905
     \hvxmp@dc@schema
906
     \hyxmp@photoshop@schema
907
     \hyxmp@photometa@schema
908
     \hyxmp@xmp@basic@schema
909
910
     \hyxmp@pdfa@id@schema
     \hyxmp@mm@schema
     \hyxmp@add@to@xml{%
913 ___</rdf:RDF>^^J%
914 </x:xmpmeta>^^J%
915 \hyxmp@padding
916 <?xpacket end="w"?>^^J%
917
    }%
918 }
```

3.6 Embedding the XMP packet

The PDF specification says that "a metadata stream may be attached to a document through the Metadata entry in the document catalogue" [3] so that's what we do here.

\hyxmp@embed@packet \hyxmp@driver

Determine which hyperref driver is in use and invoke the appropriate embedding function.

919 \newcommand*{\hyxmp@embed@packet}{%

```
\hyxmp@construct@packet
920
     \def\hyxmp@driver{hpdftex}%
921
     \ifx\hyxmp@driver\Hy@driver
922
        \hyxmp@embed@packet@pdftex
923
     \else
924
925
        \def\hyxmp@driver{hdvipdfm}%
926
        \ifx\hyxmp@driver\Hy@driver
          \hyxmp@embed@packet@dvipdfm
927
        \else
928
         \def\hyxmp@driver{hxetex}%
929
         \ifx\hyxmp@driver\Hy@driver
930
            \hyxmp@embed@packet@xetex
931
932
         \else
            \@ifundefined{pdfmark}{%
933
              \PackageWarningNoLine{hyperxmp}{%
934
                Unrecognized hyperref driver '\Hy@driver'.\MessageBreak
935
                \jobname.tex's XMP metadata will *not* be\MessageBreak
936
                embedded in the resulting file}%
937
938
           }{%
939
              \hyxmp@embed@packet@pdfmark
           }%
940
941
         \fi
       \fi
942
     \fi
943
944 }
```

3.6.1 Embedding using pdfT_EX

\hyxmp@embed@packet@pdftex E

Embed the XMP packet using pdfT_EX primitives.

```
945 \newcommand*{\hyxmp@embed@packet@pdftex}{%
946
     \bgroup
947
       \pdfcompresslevel=0
948
       \immediate\pdfobj stream attr {%
         /Type /Metadata
949
         /Subtype /XML
950
       }{\hyxmp@xml}%
951
952
       \pdfcatalog {/Metadata \the\pdflastobj\space 0 R}%
953
     \egroup
954 }
```

3.6.2 Embedding using any pdfmark-based backend

\hyxmp@embed@packet@pdfmark

Embed the XMP packet using hyperref's \pdfmark command. I believe \pdfmark is used by the dvipdf, dvipsone, dvips, dviwindo, nativepdf, pdfmark, ps2pdf, textures, and vtexpdfmark options to hyperref but I've tested only a few of those.

```
955 \newcommand*{\hyxmp@embed@packet@pdfmark}{%

956 \pdfmark{%

957 pdfmark=/NamespacePush

958 }%
```

```
\pdfmark{%
959
960
       pdfmark=/OBJ,
       Raw={/_objdef \string{hyxmp@Metadata\string} /type /stream}%
961
962
     \pdfmark{%
963
964
       pdfmark=/PUT,
965
       Raw={\string{hyxmp@Metadata\string}
966
          2 dict begin
967
            /Type /Metadata def
            /Subtype /XML def
968
            {\tt currentdict}
969
970
          end
       }%
971
972
     }%
     \pdfmark{%
973
       pdfmark=/PUT,
974
       Raw={\string{hyxmp@Metadata\string} (\hyxmp@xml)}%
975
     }%
976
977
     \pdfmark{%
978
       pdfmark=/Metadata,
       Raw={\string{Catalog\string} \string{hyxmp@Metadata\string}}%
979
     }%
980
981
     \pdfmark{%
       pdfmark=/NamespacePop
982
983
984 }
```

3.6.3 Embedding using dvipdfm

 $\verb|\hyxmp@embed@packet@dvipdfm||$

Embed the XMP packet using dvipdfm-specific \special commands. Note that dvipdfm rather irritatingly requires us to count the number of characters in the \hyxmp@xml stream ourselves.

```
985 \newcommand*{\hyxmp@embed@packet@dvipdfm}{%
      \hyxmp@string@len{\hyxmp@xml}%
986
987
      \special{pdf: object @hyxmp@Metadata
988
          /Type /Metadata
989
990
          /Subtype /XML
          /Length \the\@tempcnta
991
992
        stream^^J\hyxmp@xml endstream%
993
994
      \special{pdf: docview
995
996
997
          /Metadata @hyxmp@Metadata
998
        >>
      }%
999
1000 }
```

\hyxmp@string@len Set \@tempcnta to the number of characters in a given string (#1). The approach is

first to tally the number of space characters then to tally the number of non-space characters. While this is rather sloppy I haven't found a better way to achieve the same effect, especially given that all of the characters in #1 have already been assigned their category codes.

```
1001 \newcommand*{\hyxmp@string@len}[1]{%
1002 \@tempcnta=0
1003 \expandafter\hyxmp@count@spaces#1 {} %
1004 \expandafter\hyxmp@count@non@spaces#1{}%
1005 }
```

\hyxmp@count@spaces

Count the number of spaces in a given string. We rely on the built-in pattern matching of TEX's \def primitive to pry one word at a time off the head of the input string.

```
1006 \def\hyxmp@count@spaces#1 {%
1007 \def\hyxmp@one@token{#1}%
1008 \ifx\hyxmp@one@token\@empty
1009 \advance\@tempcnta by -1
1010 \else
1011 \advance\@tempcnta by 1
1012 \expandafter\hyxmp@count@spaces
1013 \fi
1014 }
```

\hyxmp@count@non@spaces

Count the number of non-spaces in a given string. Ideally, we'd count both spaces and non-spaces but TEX won't bind #1 to a space character (category code 10). Hence, in each iteration, #1 is bound to the next non-space character only.

```
1015 \newcommand*{\hyxmp@count@non@spaces}[1]{%
1016 \def\hyxmp@one@token{#1}%
1017 \ifx\hyxmp@one@token\@empty
1018 \else
1019 \advance\@tempcnta by 1
1020 \expandafter\hyxmp@count@non@spaces
1021 \fi
1022 }
```

3.6.4 Embedding using X_HT_EX

\hyxmp@embed@packet@xetex

Embed the XMP packet using xdvipdfmx-specific \special commands. I don't know how to tell xdvipdfmx always to leave the Metadata stream uncompressed, so the XMP metadata is likely to be missed by non-PDF-aware XMP viewers.

```
1023 \newcommand*{\hyxmp@embed@packet@xetex}{%
1024 \special{pdf:stream @hyxmp@Metadata (\hyxmp@xml)
1025 <<
1026 /Type /Metadata
1027 /Subtype /XML
1028 >>
1029 }%
1030 \special{pdf:put @catalog
```

3.7 Final clean-up

Having saved the category code of "" at the start of the package code (Section 3.1), we now restore that character's original category code.

1036 \catcode'\"=\hyxmp@dq@code

4 Future Work

Help wanted Ideally, \mmpquote should automatically replace all commas with \mmpcomma. Unfortunately, my TeX skills are insufficient to pull that off. If you know a way to make \mmpquote{Hello, world} work with both Unicode and non-Unicode encodings and with all TeX engines (pdfTeX, LuaTeX, XeTeX, etc.), please send me a code patch.

A Sample XMP packet

The following is an example of a complete XMP packet as may be produced by hyperxmp. This packet corresponds to the metadata included in the sample LATEX document presented on pages 5–6. For clarity, metadata values, either specified explicitly by the document or introduced automatically by hyperxmp, are colored blue.

```
<?xpacket begin="\357\273\277" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="3.1-702">
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      <rdf:Description rdf:about=""
           xmlns:pdf="http://ns.adobe.com/pdf/1.3/">
           <pdf:Keywords>
             energy quanta, Hertz effect, quantum physics
           </pdf:Keywords>
           <pdf:Producer>pdfTeX-1.40.10</pdf:Producer>
      </rdf:Description>
      <rdf:Description rdf:about=""
           xmlns:xmpRights="http://ns.adobe.com/xap/1.0/rights/">
         <xmpRights:Marked>True</xmpRights:Marked>
         <xmpRights:WebStatement>
           http://creativecommons.org/licenses/by-nc-nd/3.0/
         </xmpRights:WebStatement>
      </rdf:Description>
```

```
<rdf:Description rdf:about=""
     xmlns:dc="http://purl.org/dc/elements/1.1/">
   <dc:format>application/pdf</dc:format>
  <dc:title>
     <rdf:Alt>
        <rdf:li xml:lang="en">
           On a heuristic viewpoint concerning the production and
           transformation of light
         </rdf:li>
        <rdf:li xml:lang="x-default">
           On a heuristic viewpoint concerning the production and
           transformation of light
         </rdf:li>
     </rdf:Alt>
  </dc:title>
   <dc:description>
      <rdf:Alt>
         <rdf:li xml:lang="en">photoelectric effect</rdf:li>
         <rdf:li xml:lang="x-default">photoelectric effect</rdf:li>
     </rdf:Alt>
  </dc:description>
   <dc:rights>
      <rdf:Alt>
         <rdf:li xml:lang="en">
           Copyright (C) 1905, Albert Einstein
         </rdf:li>
         <rdf:li xml:lang="x-default">
           Copyright (C) 1905, Albert Einstein
        </rdf:li>
      </rdf:Alt>
  </dc:rights>
   <dc:creator>
     <rdf:Seq>
         <rdf:li>Albert Einstein</rdf:li>
      </rdf:Seq>
   </dc:creator>
   <dc:subject>
      <rdf:Bag>
         <rdf:li>energy quanta</rdf:li>
        <rdf:li>Hertz effect</rdf:li>
         <rdf:li>quantum physics</rdf:li>
     </rdf:Bag>
  </dc:subject>
   <dc:date>
     <rdf:Seq>
         <rdf:li>1905-03-17</rdf:li>
```

```
</rdf:Seq>
   </dc:date>
   <dc:language>
      <rdf:Bag>
         <rdf:li>en</rdf:li>
      </rdf:Bag>
   </dc:language>
   <dc:source>einstein.tex</dc:source>
</rdf:Description>
<rdf:Description rdf:about=""
     xmlns:photoshop="http://ns.adobe.com/photoshop/1.0/">
   <photoshop:AuthorsPosition>
    Technical Assistant, Level III
   </photoshop:AuthorsPosition>
   <photoshop:CaptionWriter>Scott Pakin</photoshop:CaptionWriter>
</rdf:Description>
<rdf:Description rdf:about=""
   xmlns:pdfaExtension="http://www.aiim.org/pdfa/ns/extension/"
   xmlns:pdfaSchema="http://www.aiim.org/pdfa/ns/schema#"
   xmlns:pdfaProperty="http://www.aiim.org/pdfa/ns/property#"
   xmlns:pdfaType="http://www.aiim.org/pdfa/ns/type#"
   xmlns:pdfaField="http://www.aiim.org/pdfa/ns/field#">
 <pdfaExtension:schemas>
   <rdf:Bag>
      <rdf:li rdf:parseType="Resource">
        <pdfaSchema:schema>IPTC Core Schema</pdfaSchema:schema>
        <pdfaSchema:namespaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/
        <pdfaSchema:prefix>Iptc4xmpCore</pdfaSchema:prefix>
        <pdfaSchema:property>
          <rdf:Seq>
            <rdf:li rdf:parseType="Resource">
              <pdfaProperty:name>CreatorContactInfo</pdfaProperty:name>
              <pdfaProperty:valueType>contactinfo</pdfaProperty:valueType>
              <pdfaProperty:category>external</pdfaProperty:category>
              <pdfaProperty:description>contact information for the document's
            </rdf:li>
          </rdf:Seq>
        </pdfaSchema:property>
        <pdfaSchema:valueType>
          <rdf:Seq>
            <rdf:li rdf:parseType="Resource">
              <pdfaType:type>contactinfo</pdfaType:type>
              <pdfaType:namespaceURI>http://iptc.org/std/Iptc4xmpCore/1.0/xmlr
              <pdfaType:prefix>IptcContInfo</pdfaType:prefix>
              <pdfaType:description>contact information</pdfaType:description>
              <pdfaType:field>
```

```
<rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiAdrExtadr</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact address</pdfaField:descript</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiAdrCity</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact city</pdfaField:description</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiAdrRegion</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact region</pdfaField:description</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiAdrPcode</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact postal code</pdfaField:desc</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiAdrCtry</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact country</pdfaField:descript</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiTelWork</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact telephone number</pdfaField</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiEmailWork</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact email address</pdfaField:de</pre>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <pdfaField:name>CiUrlWork</pdfaField:name>
            <pdfaField:valueType>Text</pdfaField:valueType>
            <pdfaField:description>contact url</pdfaField:description>
          </rdf:li>
        </rdf:Seq>
      </pdfaType:field>
    </rdf:li>
  </rdf:Seq>
</pdfaSchema:valueType>
```

<rdf:Seq>

```
</rdf:li>
          </rdf:Bag>
        </pdfaExtension:schemas>
      </rdf:Description>
      <rdf:Description rdf:about=""
           xmlns:Iptc4xmpCore="http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/"
           xmlns:IptcContInfo="http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/contactinf
       <Iptc4xmpCore:CreatorContactInfo rdf:parseType="Resource">
         <IptcContInfo:CiAdrExtadr>Kramgasse 49</IptcContInfo:CiAdrExtadr>
         <IptcContInfo:CiAdrCity>Bern</IptcContInfo:CiAdrCity>
         <IptcContInfo:CiAdrPcode>3011/IptcContInfo:CiAdrPcode>
         <IptcContInfo:CiAdrCtry>Switzerland</IptcContInfo:CiAdrCtry>
         <IptcContInfo:CiTelWork>031 312 00 91</IptcContInfo:CiTelWork>
         <IptcContInfo:CiEmailWork>aeinstein@ipi.ch</IptcContInfo:CiEmailWork>
         <IptcContInfo:CiUrlWork>
           http://einstein.biz/,
          https://www.facebook.com/AlbertEinstein
         </l></l></l></l></l><
       </Iptc4xmpCore:CreatorContactInfo>
      </rdf:Description>
      <rdf:Description rdf:about=""
           xmlns:xmp="http://ns.adobe.com/xap/1.0/">
         <xmp:CreateDate>2014-09-24T23:10:19-06:00</xmp:CreateDate>
         <xmp:ModifyDate>2014-09-24T23:10:19-06:00</xmp:ModifyDate>
         <xmp:MetadataDate>2014-09-24T23:10:19-06:00</xmp:MetadataDate>
         <xmp:CreatorTool>LaTeX with hyperref package</xmp:CreatorTool>
         <xmp:BaseURL>
           http://www.ctan.org/tex-archive/macros/latex/contrib/hyperxmp/
         </xmp:BaseURL>
      </rdf:Description>
     <rdf:Description rdf:about=""
           xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/">
         <xmpMM:DocumentID>
           uuid:0595fdce-41dc-e4c4-6c418dc4ce46
         </xmpMM:DocumentID>
         <xmpMM:InstanceID>
           uuid:efd754c4-1d7f-200a-ef754ce413ea
         </xmpMM:InstanceID>
      </rdf:Description>
  </rdf:RDF>
</x:xmpmeta>
<?xpacket end="w"?>
```

References

- [1] Adobe Systems, Inc., San Jose, California. Adobe Acrobat X SDK Help, pdfmark Reference. Available from http://www.adobe.com/devnet/acrobat/documentation.html.
- [2] Adobe Systems, Inc. *PostScript Language Reference Manual*. Addison-Wesley, 2nd edition, January 1996, ISBN: 0-201-18127-4.
- [3] Adobe Systems, Inc., San Jose, California. Document Management—Portable Document Format—Part 1: PDF 1.7, July 2008. ISO 32000-1 standard document. Available from http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf.
- [4] Adobe Systems, Inc., San Jose, California. XMP Specification Part 1: Data model, Serialization, and Core Properties, April 2012. Available from http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/cc-201306/XMPSpecificationPart1.pdf.
- [5] Michael Downes. Around the bend #15, answers, 4th (last) installment. comp.text.tex newsgroup posting, January 3, 1994. Archived by Google at http://groups.google.com/group/comp.text.tex/msg/7da7643b9e8f3b48.
- [6] International Press Telecommunications Council. IPTC Photo Metadata: Core 1.1/Extension 1.1, July 2010. Revision 1. Available from http://www.iptc.org/std/photometadata/specification/ IPTC-PhotoMetadata-201007_1.pdf.
- [7] Internet Assigned Numbers Authority. Language subtag registry, January 11, 2011. Available from http://www.iana.org/assignments/language-subtag-registry.
- [8] Paul J. Leach, Michael Mealling, and Rich Salz. A Universally Unique IDentifier (UUID) URN namespace. Request for Comments 4122, Internet Engineering Task Force, Network Working Group, July 2005. Category: Standards Track. Available from http://www.ietf.org/rfc/rfc4122.txt.
- [9] PDF/A Competence Center, Berlin, Germany. TechNote 0008: Predefined XMP Properties in PDF/A-1, March 20, 2008. Available from http://www.pdfa.org/wp-content/uploads/2011/08/tn0008_predefined_xmp_properties_in_pdfa-1_2008-03-20.pdf.
- [10] PDF/A Competence Center, Berlin, Germany. TechNote 0009: XMP Extension Schemas in PDF/A-1, March 20, 2008. Available from http://www.pdfa.org/wp-content/uploads/2011/08/tn0009_xmp_extension_schemas_in_pdfa-1_2008-03-20.pdf.

[11] Misha Wolf and Charles Wicksteed. Date and time formats. Note NOTE-datetime, World Wide Web Consortium (W3C), September 15, 1997. Available from http://www.w3.org/TR/NOTE-datetime.

Change History

v1.0	v1.5	
General: Initial version 1	General: Made the XMP inclusion	
v1.1	more robust. Thanks to Heiko	
\hyxmp@construct@packet: Explic-	Oberdiek for the bug report and	
itly set the category codes of	suggested modifications 1	0
characters $\langle EF \rangle$, $\langle BB \rangle$, and	v2.0	
$\langle BF \rangle$ to "letter". Thanks to	General: Added support for the XMP	
Daniel Schömer for the bug re-	Basic schema and miscellaneous	
port 40	other bits of metadata	1
v1.2	Heiko Oberdiek's major rewrite	
General: Added support for the	of the code to better support	
XHTFX backend (xdvipdfmx) 1	native-Unicode T _E X implemen-	
Added support for the Photoshop	tations (X¬T¬X and LuaT¬X) .	1
schema 1	New \AtBeginDocument code	
Made the package compatible	from Heiko Oberdiek to properly	
with ngerman. Thanks to Tobias	encode \@pdfmetalang 1	6
Mueller for the bug report 10	\hyxmp@add@to@xml: Updated also	
v1.3	to replace commas 2	8
General: Introduced the	\hyxmp@bom: Added by Heiko	
pdfmetalang package option,	Oberdiek 3	9
which enables an author to spec-	\hyxmp@comma: Added this macro 1	7
ify the language in which he	\hyxmp@construct@packet: Modi-	
wrote the document's metadata 16	fied by Heiko Oberdiek to use	
\hyxmp@reencode: Introduced this	an appropriate BOM representa-	
macro to re-encode Unicode	tion via \hyxmp@bom 4	0
strings as 8-bit strings before	\hyxmp@crap@convert: Added by	
manipulating them into XMP	Heiko Oberdiek 2	4
schema. This change addresses a	\hyxmp@crap@test: Added by Heiko	
bug reported by Martin Münch 20	Oberdiek 2	4
v1.4	\hyxmp@dc@schema: Added support	
\hyxmp@mm@schema: Renamed the	for dc:language and dc:source \cdot . \cdot 3	3
xapMM namespace prefix to	\hyxmp@is@unicode: Added by	
xmpMM	Heiko Oberdiek 2	21
\hyxmp@rdf@dc: Included metadata	\hyxmp@list@to@xml: Modified by	
in the x-default language re-	Heiko Oberdiek to use the new	
gardless of the specified meta-	Unicode-processing macros 3	2
data language 32	\hyxmp@photoshop@schema: Simpli-	
$\verb \hyxmp@xmpRights@schema : Re-$	fied using \hyxmp@add@simple 3	5
named the xapRights names-	$\verb \hyxmp@ProcessKeyvalOptions $	
pace prefix to xmpRights 33	Added this macro 1	4

\hyxmp@reencode: Replaced with		\hyxmp@redefine@Hyp: Added this	
an empty macro by Heiko		macro	13
Oberdiek	20	v2.2	
\hyxmp@skiptorelax: Added by		General: Added support for the IPTC	
Heiko Oberdiek	24	Photo Metadata schema	. 1
\hyxmp@skipzeros: Added by Heiko		\hyxmp@iptc@extensions: Added	
Oberdiek	23	this macro to support PDF/A	
\hyxmp@SpaceOther: Added by		generation	38
Heiko Oberdiek	23	\hyxmp@list@to@lines: Added this	
\hyxmp@string: Added this macro	31	macro	36
\hyxmp@toxml: Added by Heiko	-	\hyxmp@photometa@schema: Added	-
Oberdiek	21	this macro	36
Escaped parentheses written with		\hyxmp@text@resource: Added this	00
pdfmarks to prevent dvips from		macro	39
line-wrapping the XMP packet	22	\mmcro Changed the default	55
\hyxmp@toxml@unicodetex: Added	22	from \relax to an ordinary	
by Heiko Oberdiek	22		17
	22	comma	17
\hyxmp@xetex@crap: Added by Heiko Oberdiek	23	\xmplinesep: Added this macro .	36
	23	v2.3	
\hyxmp@xmlify: Completely rewrit-		\hyxmp@iptc@extensions: Gave	.1
ten by Heiko Oberdiek to better		the Iptc4xmpCore:CreatorContact	
support Unicode-enabled T _E X	00	fields a unique pdfaType:prefix	
programs	20	to better support conversion of	
\hyxmp@xmp@basic@schema: Added		the document to PDF/A	38
this macro	35	v2.3a	
\hyxmp@xmpRights@schema:		General: Bug fix: Redefine	
Modified to include		\@pdflang as \@empty when	
xmpRights:Marked only when		hyperref has set it to \relax .	16
pdfcopyright is specified and		v2.3b	
xmpRights:WebStatement only		\XMPTruncateList: Made all defini-	
when pdflicenseurl is speci-		tions local to avoid spurious Too	
fied	33	many unprocessed floats er-	
\hyxmp@zero: Added by Heiko		rors when running with memoir	18
Oberdiek	25	v2.4	
\ifhyxmp@unicodetex: Added by		General: Added support for the	
Heiko Oberdiek	19	PDF/A Identification schema,	
$\verb \ProcessKeyvalOptions: Added \\$		as requested by Florian Bre-	
this macro	14	itwieser	. 1
\macro	17	\hyxmp@add@simple@var: Added	
\xmpquote: Added this macro	18	this macro	31
\XMPTruncateList: Added this		\hyxmp@create@uuid: Modified this	
macro	18	macro to produce a proper ver-	
v2.1		sion 4 (random or pseudoran-	
General: Enabled hyperxmp and		dom) UUID	
hyperref to be loaded in either		\hyxmp@dc@schema: Made	
order. This addresses a bug re-		dc:language a Bag instead of an	
port by Yury Donskoy	13	individual item so as to conform	
\hypersetup: Added this macro .	14	to the latest XMP specifications,	
\hyxmp@hypersetup: Added this		a detail identified by Florian	
macro	14	Breitwieser	
			_

\hyxmp@parse@time: Added this	\hyxmp@today@define: Added this
macro 29	9 macro 30
\hyxmp@parse@tz: Added this	\mmptilde: Added this macro 18
macro 30	$0 ext{ v}2.5$
\hyxmp@parse@tz@char: Added this	General: Enabled " $_$ " to work
macro 29	9 within email addresses, as re-
\hyxmp@pdf@schema: Made	quested by Leonid Sinev 1
$\verb \hyxmp@pdf@schema always in-$	\hyxmp@add@to@xml: Updated also
clude the Adobe PDF schema,	to replace underscores and to
even when empty. Florian Bre-	modify only the text being
itwieser noted that this is neces-	added, not the already-modified
sary for PDF/A-1b compliance 3	1 text
\hyxmp@pdf@to@xmp@date: Added	\hyxmp@textunderscore: Added
this macro 29	9 this macro
\hyxmp@pdfa@id@schema: Added	\hyxmp@uscore: Added this macro 18
this macro	9 v2.6
\hyxmp@today: Modified the code	General: Added support for a new
to parse the time and timezone	pdfdate key to explicitly specify
from \pdfcreationdate, as pro-	the document date (and option-
posed by Florian Breitwieser . 30	0 ally time) 1

\mathbf{Index}

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

$\mathbf{Symbols}$	\@pdfcontactcountry	\@pdflicenseurl
\ # 519, 742	. <u>47</u> , 111, 773, 792	<u>25</u> , 121, 671, 691
\& 274, 306, 741	\@pdfcontactemail	$\verb \Qpdfmetalang 31, 139,$
\@baseurl 105,717	. <u>51</u> , 112, 775, 796	141, 143, 615, 618
\@elt . $190, 636, 754, 759$	\@pdfcontactphone	\@pdfsubject 122, 658
\@elt@first <u>752</u>	. <u>49</u> , 113, 774, 795	\@pdftitle
\@elt@rest 754, <u>756</u>	\@pdfcontactpostcode	. 123, 492, 498, 657
\@firstofone 353	<u>45</u> , 114, 772, 791	\@secondoftwo 267
\@firstoftwo 265	\@pdfcontactregion .	\^ 178, 182, 210,
\@gobble 195, 350	. <u>43</u> , 115, 771, 790	512, 513, 890–892
\@pdfauthor	\@pdfcontacturl	\ 511, 513
. <u>64</u> , 106, 492, 498	<u>53</u> , 116, 776, 797	\~ 187, 344, 345
\@pdfauthortitle	\@pdfcopyright . $\underline{23}$,	
<u>27,</u> 107, 723, 731	117, 659, 675, 685	\ 310, 345, 511
\@pdfcaptionwriter .	\@pdfcreator 716	010, 010, 011
. <u>29</u> , 108, 723, 732	\@pdfdatetime	${f A}$
\@pdfcontactaddress	. <u>21</u> , 118, 144, 146	ASCII 11, 20
. <u>33</u> , 109, 769, 788	\@pdfkeywords $\underline{81}$, $\underline{119}$	\AtBeginDocument 133
\@pdfcontactcity	\@pdflang 120, 135,	\AtEndDocument 5
. <u>41</u> , 110, 770, 789	136, 138, 141, 663	\AtEndDvi 8

atenddvi 10 Author 6, 18	H	\hyxmp@concated@metadata
Autilor 0, 18	\Hy@driver 4,	
В	922, 926, 930, 935 hyperref 1,	\hyxmp@construct@packet $$ $896, 920$
Bag 51	3-6, 10, 11, 13,	\hyxmp@count@non@spaces
baseurl (option) 3, 9, 10, 35	15, 16, 31, 40, 41, 51	$1004, \underline{1015}$
вом 39, 50	\hypersetup $\dots \dots 98$ hyperxmp $\dots 1-7$,	\hyxmp@count@spaces 1003, 1006
\mathbf{C}	9–13, 15–17, 20,	\hyxmp@crap@convert
CiAdrCity 2, 36	25, 30, 31, 44, 51	336, 370
CiAdrCtry 2, 37	\hyxmp@@is@unicode . 254	\hyxmp@crap@result .
CiAdrExtadr 2, 36	\hyxmp@add@simple	
CiAdrPcode 2, 36	582, <u>588</u> ,	\hyxmp@crap@test 333, 358
CiAdrRegion 2, 36	$664, 691, 7\overline{13}$	\hyxmp@create@uuid .
CiEmailWork 2, 37	717, 731, 732,	<u>467</u> , 495, 501
CiTelWork 2, 37	789–792, 876, 877	\hyxmp@dc@schema 651 , 906
CiUrlWork 2, 37	\hyxmp@add@simple@var	\hyxmp@def@DocumentID
CIOTIVOIR 2, 37	579, 580, <u>598</u>	
D	\hyxmp@add@to@xml	\hyxmp@def@InstanceID
Date 30		
\day 562, 563, 565	524, 575, 584,	\hyxmp@DocumentID
	593, 602, 611,	
dc:creator 2, 6, 33	617, 621, 631,	$491,703$ \hyxmp@dq@code $1,1036$
dc:date 2, 33 dc:description 3, 33	639, 645, 652,	\hyxmp@driver $\underline{3}$, $\underline{919}$
dc:format 2	665, 681, 687,	\hyxmp@embed@packet
	692, 700, 709,	
dc:language . 2, 33, 50, 51	718, 726, 735,	
dc:rights 2, 33	749, 753, 757,	\hyxmp@embed@packet@dvipdfm
dc:source 2, 33, 50	763, 781, 801,	
dc:subject 2, 33	808, 849, 862,	\hyxmp@embed@packet@pdfmark
dc:title 3, 33	872, 878, 898, 912	
\define@key 22,	\hyxmp@append@hex	\hyxmp@embed@packet@pdftex 923, 945
24, 26, 28, 30, 32,	. 440, 459–461, 465	
34, 42, 44, 46, 48,	\hyxmp@append@hex@iii	\hyxmp@embed@packet@xetex
50, 52, 54, 64, 81	. 458, 464, 474, 485	0.00000000000000000000000000000000000
dvipdf (option) 41 dvipdfm 42	\hyxmp@append@hex@iv	
dvips (option) 42	463, 469,	\hyxmp@hash
	470, 472, 487–489	. <u>518</u> , 811–814, 902
dvips 6, 22, 51	\hyxmp@at@end \dots 3, 148	\hyxmp@Hyp@pdfauthor 58
dvipsone (option) 41	\hyxmp@big@prime	
dviwindo (option) 41	. $\underline{414},417,427,437$ \hyxmp@big@prime@ii	$\frac{1}{2}$
${f E}$		\hyxmp@hypersetup 98
\EdefEscapeHex 233, 246	\hyxmp@bom 883 , 898	\hyxmp@InstanceID
\EdefUnescapeHex 250	\hyxmp@comma	
\EdefUnescapeString 220	$35, 65, 82, \underline{177}$	\hyxmp@iptc@extensions
ETX 17, 20	\hyxmp@commas@to@list	780, <u>807</u>
	. <u>161</u> , 192, 637, 761	\hyxmp@is@unicode
${f G}$	\hyxmp@commas@to@list@i	$\dots 222, 239, \underline{254}$
Ghostscript 6		$\verb \hyxmp@legal \dots \dots \underline{670}$

\hyxmp@list	\hyxmp@redefine@Hyp	\hyxmp@x@default
. 637, 643, 761, 762		. 139, <u>573</u> , 615, 622
\hyxmp@list@to@lines	$\verb \hyxmp@reencode \dots 215 $	\hyxmp@xetex@crap
. <u>745</u> , 788, 795–797	\hyxmp@rights	227, <u>326</u>
\hyxmp@list@to@xml .	. 670, 673, 677, 679	\hyxmp@xml 515,
<u>628</u> , 660–663	\hyxmp@seed@rng	<u>522, 896,</u> 951,
\hyxmp@mm@schema 697, 911	$\dots \underline{416}, 493, 499$	975, 986, 993, 1024
\hyxmp@modulo@a $\frac{300}{408}$,	\hyxmp@seed@rng@i	\hyxmp@xmlified
427, 437, 443, 478		. <u>218,</u> 594, 603,
\hyxmp@new@xml 514, 515	\hyxmp@seed@string .	618, 622, 637, 761
\hyxmp@num 370	. 492, 493, 498, 499	\hyxmp@xmlify
\hyxmp@one@token	\hyxmp@set@rand@num	. 143, 218, 592,
· -	$\frac{433}{441}$, 476	601, 610, 636, 760
$\underline{416}, \underline{420}, 1007,$	\hyxmp@skiptorelax .	\hyxmp@xmp@basic@schema
1008, 1016, 1017	363, 369	
$\label{eq:local_problem} $$ \ \ \underline{522}, 915 $$	\hyxmp@skipzeros 321	\hyxmp@xmpRights@schema
\hyxmp@parse@time	\hyxmp@SpaceOther	
536, <u>538</u>		\hyxmp@zero 379,
\hyxmp@parse@tz	\hyxmp@string <u>588</u>	386, 393, 399, <u>404</u>
$\dots 545, 548, \underline{552}$	\hyxmp@string@len	500, 555, 555, <u>404</u>
$\mbox{\hyxmp@parse@tz@char}$	986, <u>1001</u>	I
	\hyxmp@sublist	IETF 5
\hyxmp@pdf@schema	. 166, 167, 170, 171	\ifHy@pdfa 871
	\hyxmp@temp@list <u>190</u>	\ifhyxmp@unicodetex
$\verb \hyxmp@pdf@to@xmp@date \\$	\hyxmp@temp@str 190	<u>209</u> , 221, 884
	\hyxmp@text	ifxetex
\hyxmp@pdfa@id@schema	. 218, 296, 326, 370	\ifxetex 226
870, 910	\hyxmp@text@resource	Info 6
$\mbox{hyxmp@pdfauthor}$	841–848, 861	intcalc
	\hyxmp@textunderscore	\intcalcDiv 375, 382, 389
\hyxmp@pdfkeywords .		\intcalcMod 377, 384, 391
	\hyxmp@today 146,	IPTC . 7, 12, 28, 36–38, 51
\hyxmp@pdfstringdef	498, 556, 558,	Iptc4xmpCore:CreatorContactInfo
$\underline{15}, 22, 24, 26,$	560, 563, 565,	2, 36, 37, 51
28, 30, 32, 37, 42,	<u>568</u> , 662, 713–715	ISO
44, 46, 48, 50, 52, 54	\hyxmp@today@define	150
\hyxmp@photometa@data	555, 569	J
$$ $\underline{767}$	\hyxmp@toxml $\overline{248}$, $\overline{271}$	\jobname 127, 154,
\hyxmp@photometa@schema	\hyxmp@toxml@unicodetex	492, 498, 664, 936
$$ $\underline{767}$, 908		, , ,
\hyxmp@photoshop@data	\hyxmp@trimb $203, \overline{206}$	K
$ \underbrace{722} $	\hyxmp@trimc 206, <u>207</u>	Keywords 6, 31
$\verb \hyxmp@photoshop@schema $	\hyxmp@trimspaces	\KV@Hyp@pdfauthor 64
$$ $\underline{722}$, 907		\KV@Hyp@pdfkeywords 81
\hyxmp@ProcessKeyvalOptio	n\homeghyxmp@try 326	kvoptions 11, 14
<u>93</u>	\hyxmp@unicodetexfalse	
$\verb \hyxmp@rand@num = \underline{433},$	<u>209</u>	${f L}$
442, 477, 494, 500	\hyxmp@unicodetextrue	LF 36
\hyxmp@rdf@dc	<u>209</u>	$\text{Lual} \text{T}_{\text{E}} \text{X} \dots 6, 9$
$\dots \underline{607}, 657-659$	\hyxmp@uscore $17, \underline{181}$	LuaT _E X 19, 22, 44, 50

${f M}$	\pdfcreationdate 571	\StringEncodingConvert
memoir 51	pdfdate (option) . 4, 5, 52	$\dots \dots 223,$
$Metadata \dots 6, 40, 43$	PDFDocEncoding . 13, 20	229, 240, 243, 338
\month 557, 558, 560	pdfescape	Subject 6
	$pdfkeywords\ (\mathrm{option})$.	
N 11 10 00	3, 9, 10, 13, 33	T
NAK	pdflang (option)	T _E X 19, 21,
nativepdf (option) \dots 41 \newif $\dots \dots 209$	3, 5, 10, 16, 33	22, 25, 34, 43, 44, 51
\next 165, 420	\pdflastobj 952	Text
ngerman 10, 50	pdfl\(\text{PT}_E\)X 3, 6	\textunderscore
\number 373, 375, 377,	pdflicenseurl (option) .	16, 17, 19 textures (option) 41
382, 384, 389, 391	\dots 4, 5, 9, 33, 51 pdfmark (option) \dots 41	Title 6
, , , ,	\pdfmark (option) 41	Title
P	963, 973, 977, 981	${f U}$
\PackageWarningNoLine	pdfmetalang (option) 4, 5	Unicode 10, 11, 19–
\dots 126, 153, 934	\pdfminorversion 582	24, 32, 36, 39, 44, 50
PDF \cdot 1-3, 5-9, 16, 17,	\pdfobj 948	unicode (option) 10
22, 27–31, 40, 43, 52	pdfproducer (option) 3, 10	URL $2, 4, 5,$
PDF/A 3,	\pdfstringdef 18	9, 11, 13, 33–35, 37
28, 30, 38, 39, 51, 52	pdfsubject (option)	UTF-16BE 21
pdf:Keywords 2, 30, 31 pdf:PDFVersion 3, 30	3, 10, 33	UTF-32BE 20
pdf:Producer 3, 30, 31	pdfT _E X 10, 21, 41, 44	UTF-8 20, 21
pdfaid:conformance 3	pdftitle (option) 4, 10, 33	UUID $25, 27, 51$
•	photoshop:AuthorsPosition	
pdfaid:part 3		
r · · · r · · · · · · · · · · · · · · ·	3, 35	V
pdfaType:prefix 51 pdfauthor (option)	photoshop:CaptionWriter	\vfuzz 207
pdfaType:prefix 51	$\begin{array}{cccc} & \dots & 3,\ 35 \\ \text{photoshop:CaptionWriter} & & 2,\ 35 \end{array}$	
$\begin{array}{ccccc} pdfaType:prefix & \dots & 51 \\ pdfauthor & (\mathrm{option}) & \dots \\ & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle & (\mathrm{option}) & 4, 9 \end{array}$	$\begin{array}{cccc} & \dots & \dots & 3,35 \\ \text{photoshop:CaptionWriter} & & 2,35 \\ \text{PI} & \dots & & 28 \\ \end{array}$	$\label{eq:continuous_vfuzz} $$\operatorname{vfuzz} \ \dots \ \dots \ 207$$ $$\operatorname{vtexpdfmark} \ (\operatorname{option}) \ \dots \ 41$$ $$$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccc} & \dots & 3,35 \\ \text{photoshop:CaptionWriter} & & 2,35 \\ \text{PI} & \dots & 28 \\ \text{ProcessKeyvalOptions} \end{array}$	$\begin{tabular}{lll} $\tt vfuzz & \dots & \dots & 207 \\ $\tt vtexpdfmark (option) & \dots & 41 \\ & & X \\ \end{tabular}$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{lllllllllllllllllllllllllllllllllll$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{lllllllllllllllllllllllllllllllllll$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccc} & 3, 35 \\ \text{photoshop:CaptionWriter} & 2, 35 \\ \text{PI} & 28 \\ \text{ProcessKeyvalOptions} & \underline{93} \\ \text{Producer} & 31 \\ \text{ps2pdf (option)} & 41 \\ \end{array}$	$\begin{tabular}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccc} & 3, 35 \\ \text{photoshop:CaptionWriter} & 2, 35 \\ \text{PI} & 28 \\ \text{ProcessKeyvalOptions} & \underline{93} \\ \text{Producer} & 31 \\ \text{ps2pdf (option)} & 41 \\ \end{array}$	$\begin{tabular}{cccccccccccccccccccccccccccccccccccc$
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
pdfaType:prefix 51 pdfauthor (option)	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
pdfaType:prefix 51 pdfauthor (option)	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
pdfaType:prefix 51 pdfauthor (option)	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor \ (option) & \dots & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle \ (option) & 4, 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & \dots & 947 \\ pdfcontactaddress \ (option) & \dots & 4, 7, 8 \\ pdfcontactity \ (option) & \dots & 4 \\ pdfcontactity \ (option) & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactemail \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactemail \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactemail \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactemail \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactemail \ (option) & \dots & \dots & \dots \\ pdfcontactemail \ (option) & \dots & \dots &$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor & (option) & \dots \\ & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle & (option) & 4, 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & \dots & 947 \\ pdfcontactaddress & (option) & \dots & 4, 7, 8 \\ pdfcontactcity & (option) & \dots & 4 \\ pdfcontactcountry & (option) & \dots & 4 \\ pdfcontactcountry & (option) & \dots & 4 \\ pdfcontactemail & (option) & \dots & 4 \\ pdfcontactcphone & (option) & \dots & 4 \\ pdfcontactphone $	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor & (option) & \dots \\ & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle & (option) & 4, 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & 947 \\ pdfcontactaddress & (option) & \dots & 4, 7, 8 \\ pdfcontactcity & (option) & \dots & 4 \\ pdfcontactcity & (option) & \dots & 4 \\ pdfcontactcountry & (option) & \dots & \dots & 4 \\ pdfcontactemail & (option) & \dots & \dots & 4 \\ pdfcontactphone & (option) & \dots & \dots & 4 \\ pdfcontactphone & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) & \dots & \dots & \dots \\ pdfcontactpostcode & (option) &$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor & (option) & \dots \\ & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle & (option) & 4, 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & \dots & 947 \\ pdfcontactaddress & (option) & \dots & 4, 7, 8 \\ pdfcontactcity & (option) & \dots & 4 \\ pdfcontactcity & (option) & \dots & 4 \\ pdfcontactemail & (option) & \dots & \dots & 4 \\ pdfcontactemail & (option) & \dots & \dots & 4 \\ pdfcontactphone & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactpostcode & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & \dots & 4 \\ pdfcontactregion & (option) & \dots & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfcontactregion & (option) & \dots & \dots & \dots \\ pdfconta$	3, 35 photoshop:CaptionWriter	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor \ (option) & \dots & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle \ (option) & 4, 9 \\ pdfcaptionwriter & (option) & 4, 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & 947 \\ pdfcontactaddress & (option) & \dots & 4, 7, 8 \\ pdfcontactcity \ (option) & \dots & 4 \\ pdfcontactcountry \ (option) & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactphone \ (option) & \dots & \dots & 4 \\ pdfcontactpostcode \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ $	3, 35 photoshop:CaptionWriter 2, 35 PI 28 \ProcessKeyvalOptions 93 Producer 31 ps2pdf (option) 41 Q \Q 199, 208 R rdf:li 2 rdf:Seq 2 \renewcommand 94 \RequirePackage 7, 10-14 S \SE->pdfdoc@03 216 \SE->pdfdoc@15 217	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor \ (option) & \dots & \dots & 3, \ 9, \ 10, \ 13, \ 33 \\ pdfauthortitle \ (option) & 4, \ 9 \\ pdfcaptionwriter & (option) & 4, \ 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & 947 \\ pdfcontactaddress \ (option) & \dots & 4, \ 7, \ 8 \\ pdfcontactaddress \ (option) & \dots & 4 \\ pdfcontactcountry \ (option) & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactphone \ (option) & \dots & \dots & 4 \\ pdfcontactregion \ (option) & \dots & \dots & 4 \\ pdfcontactregion \ (option) & \dots & \dots & 4 \\ pdfcontactregion \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & \dots & 4 \\ pdfcontacturl \ (option) & \dots & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontacturl \ (option) & \dots & \dots & \dots \\ pdfcontact$	3, 35 photoshop:CaptionWriter	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
$\begin{array}{c} pdfaType:prefix & \dots & 51 \\ pdfauthor \ (option) & \dots & \dots & 3, 9, 10, 13, 33 \\ pdfauthortitle \ (option) & 4, 9 \\ pdfcaptionwriter & (option) & 4, 9 \\ pdfcaptionwriter & (option) & \dots & 4 \\ pdfcatalog & \dots & 952 \\ pdfcompresslevel & 947 \\ pdfcontactaddress & (option) & \dots & 4, 7, 8 \\ pdfcontactcity \ (option) & \dots & 4 \\ pdfcontactcountry \ (option) & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactemail \ (option) & \dots & \dots & 4 \\ pdfcontactphone \ (option) & \dots & \dots & 4 \\ pdfcontactpostcode \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & \dots & 4 \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ (option) & \dots & \dots & \dots \\ pdfcontactengion \ $	3, 35 photoshop:CaptionWriter 2, 35 PI 28 \ProcessKeyvalOptions 93 Producer 31 ps2pdf (option) 41 Q \Q 199, 208 R rdf:li 2 rdf:Seq 2 \renewcommand 94 \RequirePackage 7, 10-14 S \SE->pdfdoc@03 216 \SE->pdfdoc@15 217	$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$

xmpMM:DocumentID .	36, 39, <u>64</u> , <u>81</u> , <u>185</u>	\XMPTruncateList $\underline{190}$
$\dots \dots 2, 25, 34$	xmpRights:Marked $2, 33, 51$	
xmpMM:InstanceID	xmpRights:WebStatement	
$\dots \dots 2, 25, 34$		Y
\xmpquote	\xmptilde 186	\year 556