

The embedfile package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2006/08/16 v1.0

Abstract

This package embeds files to a PDF document. Currently pdfTeX 1.30 in PDF mode is supported only.

Contents

| | | |
|----------|---|-----------|
| 1 | Documentation | 1 |
| 1.1 | Introduction | 1 |
| 1.1.1 | Future development | 2 |
| 1.2 | User interface | 2 |
| 1.3 | Examples | 3 |
| 1.3.1 | plain-T _E X | 3 |
| 1.4 | Package dtx-attach | 3 |
| 2 | Implementation | 4 |
| 2.1 | Reload check and package identification | 4 |
| 2.2 | Tools | 5 |
| 2.3 | Check for recent pdfTeX in PDF mode | 5 |
| 2.4 | Key value definitions | 6 |
| 2.5 | Embed the file | 6 |
| 3 | Installation | 9 |
| 3.1 | Download | 9 |
| 3.2 | Bundle installation | 9 |
| 3.3 | Package installation | 9 |
| 3.4 | Refresh file name databases | 10 |
| 3.5 | Some details for the interested | 10 |
| 4 | References | 10 |
| 5 | History | 11 |
| | [2006/08/16 v1.0] | 11 |
| 6 | Index | 11 |

1 Documentation

1.1 Introduction

The PDF format ([3]) allows the inclusion of files inside the PDF document. The included files can be bound to an annotation on a page. Or they can be recorded in a sorted list of embedded files. The packages `attachfile` or `attachfile2` follow the first approach, this package uses the latter method.

1.1.1 Future development

My dream is a large package that merges the features of all these packages mentioned before:

- Files can be attached to a page.
- Files can be attached to the document.
- An easy user interface for simple, common tasks and beginners.
- An interface for the advanced users that want to setup every detail.
- Support of many drivers (pdftex, dvips, dvipdfm, ...).
- ...

However, I have not managed to take the time for this project. Instead:

- First I experimented with package `attachfile`, adding driver support, fixing bugs, The result is currently named as `attachfile2`. It uses an external script to get file properties (size, date, checksum, ...).
- In order to avoid an external program for getting basic file properties I provided a patch “EscapeAndOther” for pdfTeX that was accepted for version 1.30.
- This package closes a gap left by the packages for attaching files and allows the embedding of files to the document. Also it makes use of the new primitives of pdfTeX.

1.2 User interface

This package `embedfile` can be used with both L^AT_EX and plain-T_EX. See [subsubsection 1.3.1](#) that explains the use with plain-T_EX by an example. In L^AT_EX the package is loaded as usually. There are no options.

```
\usepackage{embedfile}
```

`\embedfile [options] {file}`

The macro `\embedfile` includes file *file* and attaches it to the PDF document. At the end of the document the sorted list of embedded files are written. Thus you can safely use `\embedfile` before `\end{document}`. Embedding files using `\AtEndDocument` will only work, if `\AtEndDocument` is called before loading the package `embedfile`.

The *options* are give as key value pairs. The following keys are supported:

filespec This allows to override the file name that appears in the PDF file. If you are using other than simple file names (8bit, path separators, ...), look into the PDF specification ([3]). There are rules how these file names must be written/encoded.

filesystem This sets the entry /FS in the file specification dictionary, see PDF specification ([3]). Example: `filesystem=URL`.

mimetype This sets the mime type ([4]) of the file, see [subsubsection 1.3.1](#) for examples and [5] for a list of officially registered types.

desc The description for the file.

stringmethod The package must convert the values of the keys `filespec` and `desc` into a PDF string. If `hyperref` is found, then its `\pdfstringdef` will be used, otherwise pdfTeX's `\pdfescapestring` is used. Value `psd` forces the use of `\pdfstringdef`, value `escape` the use of `\pdfescapestring`.

1.3 Examples

1.3.1 plain-TeX

The package can be used with plain-TeX. Because it uses package `keyval` and `\@ifnextchar`, it needs a little help from `miniltx.tex`.

If additionally package `keyval` (`graphicx`) is needed, load it first. Then package `embedfile` avoids a duplicate loading of package `keyval`.

Because plain-TeX does not provide a hook at end of the document, you have to call `\embedfilefinish` manually at the end after the last embedded file.

```
1 (*exampleplain)
2 % Load packages
3 \input miniltx
4 % \def\Gin@driver{pdftex.def}
5 % \input graphicx.sty
6 \input embedfile.sty
7 \resetatcatcode
8
9 % default setting
10 \embedfilesetup{
11   mimetype=text/plain
12 }
13
14 % Embed files
15 \embedfile[
16   filespec=example.tex,
17   desc={Source code (plain-TeX) of this example}
18 ]{embedfile-example-plain.tex}
19
20 \embedfile[
21   desc={Source of package 'embedfile'}
22 ]{embedfile.dtx}
23
24 \embedfile[
25   mimetype=application/pdf,
26   desc={Documentation of package 'embedfile'}
27 ]{embedfile.pdf}
28
29 % Some text
30 This example document contains three embedded files.
31
32 % End of document
33 \embedfilefinish % don't forget
34 \bye
35 /exampleplain)
```

1.4 Package `dtx-attach`

Package `dtx-attach` is just a small application of package `embedfile`. I am using it for the CTAN documentation of my packages in [CTAN:macros/latex/contrib/oberdiek/](https://ctan.org/ctan/latex/contrib/oberdiek/). It also serves as small example for the use of the package with L^AT_EX.

```
36 (*dtxattach)
37 \NeedsTeXFormat{LaTeX2e}
38 \ProvidesPackage{dtx-attach}
39 [2006/08/16 v1.0 Embed \string\jobname.dtx (H0)]
40 \RequirePackage{embedfile}[2006/08/16]
41 \embedfile[
42   stringmethod=escape,
43   mimetype=plain/text,
44   desc={LaTeX docstrip source archive for package '\jobname'}
45 ]{\jobname.dtx}
46 /dtxattach)
```

2 Implementation

47 `\package`

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```
48 \begingroup
49   \expandafter\let\expandafter\x\csname ver@embedfile.sty\endcsname
50   \ifcase 0%
51     \ifx\x\relax % plain
52     \else
53       \ifx\x\empty % LaTeX
54       \else
55         1%
56       \fi
57     \fi
58   \else
59     \expandafter\ifx\csname PackageInfo\endcsname\relax
60     \def\x#1#2{%
61       \immediate\write-1{Package #1 Info: #2.}%
62     }%
63   \else
64     \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
65   \fi
66   \x{embedfile}{The package is already loaded}%
67 \endgroup
68 \expandafter\endinput
69 \fi
70 \endgroup
```

Package identification:

```
71 \begingroup
72   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
73   \def\x#1#2#3[#4]{\endgroup
74     \immediate\write-1{Package: #3 #4}%
75     \xdef#1{#4}%
76   }%
77   \else
78     \def\x#1#2[#3]{\endgroup
79       #2[#3]}%
80     \ifx#1\relax
81       \xdef#1{#3}%
82     \fi
83   }%
84 \fi
85 \expandafter\x\csname ver@embedfile.sty\endcsname
86 \ProvidesPackage{embedfile}%
87 [2006/08/16 v1.0 embed files into PDF (H0)]

88 \edef\EmbedFileRestoreCatcodes{%
89   \catcode39 \the\catcode39 % '
90   \catcode40 \the\catcode40 % (
91   \catcode41 \the\catcode41 % )
92   \catcode47 \the\catcode47 % /
93   \catcode60 \the\catcode60 % <
94   \catcode62 \the\catcode62 % >
95   \catcode64 \the\catcode64 % @
96   \catcode91 \the\catcode91 % [
97   \catcode93 \the\catcode93 % ]
98   \catcode96 \the\catcode96 % '
99 }
100 \catcode39 12 % '
101 \catcode40 12 % (
```

```

102 \catcode41 12 % )
103 \catcode47 12 % /
104 \catcode60 12 % <
105 \catcode62 12 % >
106 \catcode64 11 % @
107 \catcode91 12 % [
108 \catcode93 12 % ]
109 \catcode96 12 % '

```

2.2 Tools

`\EmbedFile@PackageError`

```

110 \begingroup\expandafter\expandafter\expandafter\endgroup
111 \expandafter\ifx\csname PackageError\endcsname\relax
112   \def\EmbedFile@PackageError#1#2{%
113     \errhelp{#2}%
114     \errmessage{Package embedfile Error: #1.}%
115   }%
116 \else
117   \def\EmbedFile@PackageError#1#2{%
118     \begingroup
119       \let\on@line\empty
120       \PackageError{embedfile}{#1}{#2}%
121     \endgroup
122   }%
123 \fi

```

`\EmbedFile@RequirePackage`

```

124 \begingroup\expandafter\expandafter\expandafter\endgroup
125 \expandafter\ifx\csname RequirePackage\endcsname\relax
126   \def\EmbedFile@RequirePackage#1#2{%
127     \expandafter\ifx\csname #2\endcsname\relax
128       \input #1.sty\relax
129     \fi
130   }%
131 \else
132   \def\EmbedFile@RequirePackage#1#2{%
133     \expandafter\ifx\csname #2\endcsname\relax
134       \RequirePackage{#1}%
135     \fi
136   }%
137 \fi

```

2.3 Check for recent pdfTeX in PDF mode

Load package ifpdf and check mode.

```

138 \EmbedFile@RequirePackage{ifpdf}{ifpdf}
139 \ifpdf
140 \else
141   \EmbedFile@PackageError{%
142     Missing pdfTeX in PDF mode%
143   }{%
144     Currently other drivers are not supported. Package loading is aborted.%
145   }%
146   \EmbedFileRestoreCatcodes
147   \expandafter\endinput
148 \fi

```

Check version.

```

149 \begingroup\expandafter\expandafter\expandafter\endgroup
150 \expandafter\ifx\csname pdffilesize\endcsname\relax
151   \EmbedFile@PackageError{%

```

```

152     Unsupported pdfTeX version%
153 }{%
154     At least version 1.30 is necessary. Package loading is aborted.%
155 }%
156 \EmbedFileRestoreCatcodes
157 \expandafter\endinput
158 \fi

```

2.4 Key value definitions

```

159 \EmbedFile@RequirePackage{keyval}{define@key}
160 \def\EmbedFile@DefineKey#1#2{%
161     \define@key{EmbedFile}{#1}{%
162         \expandafter\def\csname EmbedFile@#1\endcsname{##1}%
163     }%
164     \expandafter\def\csname EmbedFile@#1\endcsname{#2}%
165 }

```

Subtype of the embedded file (optional).

```
166 \EmbedFile@DefineKey{mimetype}{}
```

File specification string.

```
167 \EmbedFile@DefineKey{filespec}{\EmbedFile@file}
```

File system (optional).

```
168 \EmbedFile@DefineKey{filesystem}{}
```

Description (optional).

```
169 \EmbedFile@DefineKey{desc}{}
```

Method for converting text to PDF strings.

```

170 \EmbedFile@DefineKey{stringmethod}{%
171     \ifx\pdfstringdef\@undefined
172         escape%
173     \else
174         \ifx\pdfstringdef\relax
175             escape%
176         \else
177             psd%
178         \fi
179     \fi
180 }

```

\embedfilessetup

```

181 \def\embedfilessetup{%
182     \setkeys{EmbedFile}%
183 }

```

2.5 Embed the file

\embedfile

```

184 \def\embedfile{%
185     \@ifnextchar[\EmbedFile@embedfile{\EmbedFile@embedfile[]}%
186 }

```

\EmbedFile@embedfile

```

187 \def\EmbedFile@embedfile[#1]#2{%
188     \ifEmbedFile@finished
189         \EmbedFile@PackageError{%
190             \string\embedfile\space after \string\embedfilefinish
191         }{%
192             The list of embedded files is already written.%
193         }%
194     \else
195         \begingroup

```

```

196 \def\EmbedFile@file{#2}%
197 \setkeys{EmbedFile}{#1}%
198 \expandafter\ifx\expandafter\\pdffilesize{\EmbedFile@file}\\%
199 \EmbedFile@PackageError{%
200   File '\EmbedFile@file' not found%
201 }{%
202   The unknown file is not embedded.%
203 }%
204 \else
205 \EmbedFile@convert\EmbedFile@filespec\EmbedFile@@filespec
206 \ifx\EmbedFile@desc\empty
207 \let\EmbedFile@@desc\empty
208 \else
209 \EmbedFile@convert\EmbedFile@desc\EmbedFile@@desc
210 \fi
211 \immediate\pdfobj stream attr{%
212   /Type/EmbeddedFile%
213   \ifx\EmbedFile@mimetype\empty
214   \else
215     /Subtype/\pdfescapename{\EmbedFile@mimetype}%
216   \fi
217   /Params<<%
218     /ModDate(\pdffilemoddate{\EmbedFile@file})%
219     /Size \pdffilesize{\EmbedFile@file}%
220     /Checksum<\pdfmdfivesum file{\EmbedFile@file}>%
221   >>%
222 }file{\EmbedFile@file}\relax
223 \immediate\pdfobj{%
224   <<%
225     /Type/Filespec%
226     \ifx\EmbedFile@filesystem\empty
227     \else
228       /FS/\pdfescapename{\EmbedFile@filesystem}%
229     \fi
230     /F(\EmbedFile@@filespec)%
231     \ifx\EmbedFile@@desc\empty
232     \else
233       /Desc(\EmbedFile@@desc)%
234     \fi
235     /EF<<%
236       /F \the\pdflastobj\space 0 R%
237     >>%
238   >>%
239 }%
240 \EmbedFile@add{%
241   \EmbedFile@@filespec
242 }{\the\pdflastobj\space 0 R}%
243 \fi
244 \endgroup
245 \fi
246 }

```

\EmbedFile@convert

```

247 \def\EmbedFile@convert#1#2{%
248   \ifnum\pdfstrcmp{\EmbedFile@stringmethod}{psd}=0 %
249   \pdfstringdef\EmbedFile@temp{#1}%
250   \let#2\EmbedFile@temp
251   \else
252     \edef#2{\pdfescapestring{#1}}%
253   \fi
254 }

255 \global\let\EmbedFile@list\empty

```

`\EmbedFile@add` Sorting is done by the insertion sort algorithm. Probably the sorting could be done more reliable. However, the PDF specification is not too clear to me regarding precise sorting rules (how to deal with different encodings, escaped characters, ...).

```

256 \def\EmbedFile@add#1#2{%
257   \begingroup
258   \edef\key{\pdfescapehex{#1}}%
259   \ifx\EmbedFile@list\empty
260     \xdef\EmbedFile@list{\noexpand\do{\key}{#2}}%
261   \else
262     \def\do##1##2{%
263       \ifnum\pdfstrcmp{##1}{\key}>0 %
264         \edef\x{%
265           \toks@{%
266             \the\toks@%
267             \noexpand\do{\key}{#2}%
268             \noexpand\do{##1}{##2}%
269           }%
270         }%
271         \x
272       \def\do####1####2{%
273         \toks@\expandafter{\the\toks@\do{####1}{####2}}%
274       }%
275       \def\stop{%
276         \xdef\EmbedFile@list{\the\toks@}%
277       }%
278     \else
279       \toks@\expandafter{\the\toks@\do{##1}{##2}}%
280     \fi
281   }%
282   \def\stop{%
283     \xdef\EmbedFile@list{\the\toks@\noexpand\do{\key}{#2}}%
284   }%
285   \toks@{}%
286   \EmbedFile@list\stop
287 \fi
288 \endgroup
289 }

```

290 \newif\ifEmbedFile@finished

`\embedfilefinish`

```

291 \def\embedfilefinish{%
292   \ifEmbedFile@finished
293     \EmbedFile@PackageError{%
294       Too many invocations of \string\embedfilefinish
295     }{%
296       The list of embedded files is already written.%
297     }%
298   \else
299     \ifx\EmbedFile@list\empty
300     \else
301       \global\EmbedFile@finishedtrue
302       \begingroup
303       \def\do##1##2{%
304         <##1>##2%
305       }%
306       \immediate\pdfobj{%
307         <<%
308         /Names[\EmbedFile@list]%
309         >>%
310       }%

```



```

311         \pdfnames{%
312             /EmbeddedFiles \the\pdflastobj\space 0 R%
313         }%
314     \endgroup
315     \fi
316 \fi
317 }

318 \begingroup\expandafter\expandafter\expandafter\endgroup
319 \expandafter\ifx\csname AtEndDocument\endcsname\relax
320 \else
321     \AtEndDocument{\embedfilefinish}%
322 \fi

323 \EmbedFileRestoreCatcodes
324 \endpackage

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/embedfile.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/embedfile.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

3.2 Bundle installation

Unpacking. Unpack the oberdiek-tds.zip in the TDS tree (also known as texmf tree) of your choice. Example (linux):

```
unzip oberdiek-tds.zip -d ~/texmf
```

Script installation. Check the directory TDS:scripts/oberdiek/ for scripts that need further installation steps. Package attachfile2 comes with the Perl script pdfatfi.pl that should be installed in such a way that it can be called as pdfatfi. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

3.3 Package installation

Unpacking. The .dtx file is a self-extracting docstrip archive. The files are extracted by running the .dtx through plain-TEX:

```
tex embedfile.dtx
```

¹<http://ftp.ctan.org/tex-archive/>

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

| | | |
|--|---|---|
| <code>embedfile.sty</code> | → | <code>tex/latex/oberdiek/embedfile.sty</code> |
| <code>dtx-attach.sty</code> | → | <code>tex/latex/oberdiek/dtx-attach.sty</code> |
| <code>embedfile.pdf</code> | → | <code>doc/latex/oberdiek/embedfile.pdf</code> |
| <code>embedfile-example-plain.tex</code> | → | <code>doc/latex/oberdiek/embedfile-example-plain.tex</code> |
| <code>embedfile.dtx</code> | → | <code>source/latex/oberdiek/embedfile.dtx</code> |

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

3.4 Refresh file name databases

If your \TeX distribution (`te \TeX` , `mik \TeX` , ...) relies on file name databases, you must refresh these. For example, `te \TeX` users run `texhash` or `mktextlsr`.

3.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk embedfile.pdf unpack_files output .
```

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:

plain- \TeX : Run `docstrip` and extract the files.

\LaTeX : Generate the documentation.

If you insist on using \LaTeX for `docstrip` (really, `docstrip` does not need \LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{embedfile.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf \LaTeX` :

```
pdflatex embedfile.dtx
makeindex -s gind.ist embedfile.idx
pdflatex embedfile.dtx
makeindex -s gind.ist embedfile.idx
pdflatex embedfile.dtx
```

4 References

- [1] Scott Pakin: *The attachfile package*; 2005/02/20 v1.2; [CTAN:macros/latex/contrib/attachfile/](#).
- [2] Heiko Oberdiek: *The attachfile2 package*; 2006/08/16 v2.2; [CTAN:macros/latex/contrib/oberdiek/attachfile2.pdf](#).

- [3] Adobe Systems Incorporated: *PDF Reference, Fifth Edition, Version 1.6*, November 2004; http://partners.adobe.com/public/developer/pdf/index_reference.html.
- [4] Network Working Group: RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, November 1996; <http://www.rfc-editor.org/>.
- [5] IANA (Internet Assigned Numbers Authority): *MIME Media Types*, May 2006; <http://www.iana.org/assignments/media-types/>.

5 History

[2006/08/16 v1.0]

- First public version,

6 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.

| Symbols | |
|---------------------------------|---|
| \@ifnextchar | 185 |
| \@undefined | 171 |
| \\ | 198 |
| A | |
| \AtEndDocument | 321 |
| B | |
| \bye | 34 |
| C | |
| \catcode | 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109 |
| \csname | 49, 59, 72, 85, 111, 125, 127, 133, 150, 162, 164, 319 |
| D | |
| \define@key | 161 |
| \do | 260, 262, 267, 268, 272, 273, 279, 283, 303 |
| E | |
| \embedfile . | 2, 15, 20, 24, 41, <u>184</u> , 190 |
| \EmbedFile@@desc .. | 207, 209, 231, 233 |
| \EmbedFile@@filespec .. | 205, 230, 241 |
| \EmbedFile@add | 240, <u>256</u> |
| \EmbedFile@convert | 205, 209, <u>247</u> |
| \EmbedFile@DefineKey | 160, 166, 167, 168, 169, 170 |
| \EmbedFile@desc | 206, 209 |
| \EmbedFile@embedfile | 185, <u>187</u> |
| \EmbedFile@file | 167, 196, 198, 200, 218, 219, 220, 222 |
| \EmbedFile@filespec | 205 |
| \EmbedFile@filesystem | 226, 228 |
| \EmbedFile@finishedtrue | 301 |
| \EmbedFile@list | 255, 259, 260, 276, 283, 286, 299, 308 |
| \EmbedFile@mimetype | 213, 215 |
| \EmbedFile@PackageError | 110, 141, 151, 189, 199, 293 |
| \EmbedFile@RequirePackage | 124, 138, 159 |
| \EmbedFile@stringmethod | 248 |
| \EmbedFile@temp | 249, 250 |
| \embedfilefinish ... | 33, 190, <u>291</u> , 321 |
| \EmbedFileRestoreCatcodes | 88, 146, 156, 323 |
| \embedfilesetup | 10, <u>181</u> |
| \empty | 53, 119, 206, 207, 213, 226, 231, 255, 259, 299 |
| \encsname | 127 |
| \endcsname | 49, 59, 72, 85, 111, 125, 133, 150, 162, 164, 319 |
| \endinput | 68, 147, 157 |
| \errhelp | 113 |
| \errmessage | 114 |
| G | |
| \Gin@driver | 4 |
| I | |
| \ifcase | 50 |
| \ifEmbedFile@finished . | 188, 290, 292 |
| \ifnum | 248, 263 |
| \ifpdf | 139 |
| \ifx ... | 51, 53, 59, 72, 80, 111, 125, 127, 133, 150, 171, 174, 198, 206, 213, 226, 231, 259, 299, 319 |
| \immediate | 61, 74, 211, 223, 306 |
| \input | 3, 5, 6, 128 |
| J | |
| \jobname | 39, 44, 45 |

| | | |
|-----------------------------------|-------------------------|---|
| K | | \ProvidesPackage 38, 86 |
| \key | 258, 260, 263, 267, 283 | |
| N | | R |
| \NeedsTeXFormat | 37 | \RequirePackage 40, 134 |
| \newif | 290 | \resetatcatcode 7 |
| O | | S |
| \on@line | 119 | \setkeys 182, 197 |
| P | | \space 190, 236, 242, 312 |
| \PackageError | 120 | \stop 275, 282, 286 |
| \PackageInfo | 64 | |
| \pdfescapehex | 258 | T |
| \pdfescapename | 215, 228 | \the 89, 90, 91, |
| \pdfescapestring | 252 | 92, 93, 94, 95, 96, 97, 98, 236, |
| \pdffilemoddate | 218 | 242, 266, 273, 276, 279, 283, 312 |
| \pdffilesize | 198, 219 | \toks@ 265, 266, 273, 276, 279, 283, 285 |
| \pdflastobj | 236, 242, 312 | |
| \pdfmdfivesum | 220 | W |
| \pdfnames | 311 | \write 61, 74 |
| \pdfobj | 211, 223, 306 | |
| \pdfstrcmp | 248, 263 | X |
| \pdfstringdef | 171, 174, 249 | \x 49, 51, |
| | | 53, 60, 64, 66, 73, 78, 85, 264, 271 |