

The embedfile package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/04/11 v1.1

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-TeX	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	10
3.4	Refresh file name databases	10
3.5	Some details for the interested	10
4	References	11
5	History	11
	[2006/08/16 v1.0]	11
	[2007/04/11 v1.1]	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/. 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 [2007/04/11 v1.1 Embed \string\jobname.dtx (H0)]%
40 \RequirePackage{embedfile}[2007/04/11]
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 \catcode44 12 % ,
50 \catcode45 12 % -
51 \catcode46 12 % .
52 \catcode58 12 % :
53 \catcode64 11 % @
54 \expandafter\let\expandafter\x\csname ver@embedfile.sty\endcsname
55 \ifcase 0%
56 \ifx\x\relax % plain
57 \else
58 \ifx\x\empty % LaTeX
59 \else
60 1%
61 \fi
62 \fi
63 \else
64 \expandafter\ifx\csname PackageInfo\endcsname\relax
65 \def\x#1#2{%
66 \immediate\write-1{Package #1 Info: #2.}%
67 }%
68 \else
69 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
70 \fi
71 \x{embedfile}{The package is already loaded}%
72 \endgroup
73 \expandafter\endinput
74 \fi
75 \endgroup
```

Package identification:

```
76 \begingroup
77 \catcode44 12 % ,
78 \catcode45 12 % -
79 \catcode46 12 % .
80 \catcode58 12 % :
81 \catcode64 11 % @
82 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
83 \def\x#1#2#3[#4]{\endgroup
84 \immediate\write-1{Package: #3 #4}%
85 \xdef#1{#4}%
86 }%
87 \else
88 \def\x#1#2[#3]{\endgroup
89 #2[#3]}%
90 \ifx#1\relax
91 \xdef#1{#3}%
92 \fi
93 }%
94 \fi
95 \expandafter\x\csname ver@embedfile.sty\endcsname
96 \ProvidesPackage{embedfile}%
97 [2007/04/11 v1.1 embed files into PDF (H0)]

98 \edef\EmbedFileRestoreCatcodes{%
99 \catcode39 \the\catcode39 % '
100 \catcode40 \the\catcode40 % (
101 \catcode41 \the\catcode41 % )
```

```

102 \catcode47 \the\catcode47 % /
103 \catcode60 \the\catcode60 % <
104 \catcode62 \the\catcode62 % >
105 \catcode64 \the\catcode64 % @
106 \catcode91 \the\catcode91 % [
107 \catcode93 \the\catcode93 % ]
108 \catcode96 \the\catcode96 % '
109 }
110 \catcode39 12 % '
111 \catcode40 12 % (
112 \catcode41 12 % )
113 \catcode47 12 % /
114 \catcode60 12 % <
115 \catcode62 12 % >
116 \catcode64 11 % @
117 \catcode91 12 % [
118 \catcode93 12 % ]
119 \catcode96 12 % '

```

2.2 Tools

`\EmbedFile@PackageError`

```

120 \begingroup\expandafter\expandafter\expandafter\endgroup
121 \expandafter\ifx\csname PackageError\endcsname\relax
122 \def\EmbedFile@PackageError#1#2{%
123 \errhelp{#2}%
124 \errmessage{Package embedfile Error: #1.}%
125 }%
126 \else
127 \def\EmbedFile@PackageError#1#2{%
128 \begingroup
129 \let\on@line\empty
130 \PackageError{embedfile}{#1}{#2}%
131 \endgroup
132 }%
133 \fi

```

`\EmbedFile@RequirePackage`

```

134 \begingroup\expandafter\expandafter\expandafter\endgroup
135 \expandafter\ifx\csname RequirePackage\endcsname\relax
136 \def\EmbedFile@RequirePackage#1#2{%
137 \expandafter\ifx\csname #2\endcsname\relax
138 \input #1.sty\relax
139 \fi
140 }%
141 \else
142 \def\EmbedFile@RequirePackage#1#2{%
143 \expandafter\ifx\csname #2\endcsname\relax
144 \RequirePackage{#1}%
145 \fi
146 }%
147 \fi

```

2.3 Check for recent pdfTeX in PDF mode

Load package ifpdf and check mode.

```

148 \EmbedFile@RequirePackage{ifpdf}{ifpdf}
149 \ifpdf
150 \else
151 \EmbedFile@PackageError{%
152 Missing pdfTeX in PDF mode%

```

```

153 }{%
154     Currently other drivers are not supported. Package loading is aborted.%
155 }%
156 \EmbedFileRestoreCatcodes
157 \expandafter\endinput
158 \fi

Check version.

159 \begingroup\expandafter\expandafter\expandafter\endgroup
160 \expandafter\ifx\csname pdffilesize\endcsname\relax
161     \EmbedFile@PackageError{%
162         Unsupported pdfTeX version%
163     }{%
164         At least version 1.30 is necessary. Package loading is aborted.%
165     }%
166     \EmbedFileRestoreCatcodes
167 \expandafter\endinput
168 \fi

```

2.4 Key value definitions

```

169 \EmbedFile@RequirePackage{keyval}{\define@key}

170 \def\EmbedFile@DefineKey#1#2{%
171     \define@key{EmbedFile}{#1}{%
172         \expandafter\def\csname EmbedFile@#1\endcsname{##1}%
173     }%
174     \expandafter\def\csname EmbedFile@#1\endcsname{#2}%
175 }

```

Subtype of the embedded file (optional).

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

File specification string.

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

File system (optional).

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

Description (optional).

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

Method for converting text to PDF strings.

```

180 \EmbedFile@DefineKey{stringmethod}{%
181     \ifx\pdfstringdef\@undefined
182         escape%
183     \else
184         \ifx\pdfstringdef\relax
185             escape%
186         \else
187             psd%
188         \fi
189     \fi
190 }

```

`\embedfilesetup`

```

191 \def\embedfilesetup{%
192     \setkeys{EmbedFile}%
193 }

```

2.5 Embed the file

`\embedfile`

```

194 \def\embedfile{%
195     \@ifnextchar[\EmbedFile@embedfile{\EmbedFile@embedfile[]}%
196 }

```

\EmbedFile@embedfile

```
197 \def\EmbedFile@embedfile[#1]#2{%
198   \ifEmbedFile@finished
199     \EmbedFile@PackageError{%
200       \string\embedfile\space after \string\embedfilefinish
201     }{%
202       The list of embedded files is already written.%
203     }%
204   \else
205     \begingroup
206       \def\EmbedFile@file{#2}%
207       \setkeys{EmbedFile}{#1}%
208       \expandafter\ifx\expandafter\\pdffilesize{\EmbedFile@file}\\%
209       \EmbedFile@PackageError{%
210         File '\EmbedFile@file' not found%
211       }{%
212         The unknown file is not embedded.%
213       }%
214     \else
215       \EmbedFile@convert\EmbedFile@filespec\EmbedFile@@filespec
216       \ifx\EmbedFile@desc\empty
217         \let\EmbedFile@@desc\empty
218       \else
219         \EmbedFile@convert\EmbedFile@desc\EmbedFile@@desc
220       \fi
221       \immediate\pdfobj stream attr{%
222         /Type/EmbeddedFile%
223         \ifx\EmbedFile@mimetype\empty
224         \else
225           /Subtype/\pdfescapename{\EmbedFile@mimetype}%
226         \fi
227         /Params<<%
228           /ModDate(\pdffilemoddate{\EmbedFile@file})%
229           /Size \pdffilesize{\EmbedFile@file}%
230           /Checksum<\pdfmdfivesum file{\EmbedFile@file}>%
231         >>%
232       }file{\EmbedFile@file}\relax
233       \immediate\pdfobj{%
234         <<%
235           /Type/Filespec%
236           \ifx\EmbedFile@filesystem\empty
237           \else
238             /FS/\pdfescapename{\EmbedFile@filesystem}%
239           \fi
240           /F(\EmbedFile@@filespec)%
241           \ifx\EmbedFile@@desc\empty
242           \else
243             /Desc(\EmbedFile@@desc)%
244           \fi
245           /EF<<%
246             /F \the\pdflastobj\space 0 R%
247           >>%
248         >>%
249       }%
250       \EmbedFile@add{%
251         \EmbedFile@@filespec
252       }{\the\pdflastobj\space 0 R}%
253     \fi
254   \endgroup
255 \fi
256 }
```

\EmbedFile@convert

```
257 \def\EmbedFile@convert#1#2{%
258   \ifnum\pdfstrcmp{\EmbedFile@stringmethod}{psd}=0 %
259     \pdfstringdef\EmbedFile@temp{#1}%
260     \let#2\EmbedFile@temp
261   \else
262     \edef#2{\pdfescapestring{#1}}%
263   \fi
264 }

265 \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, ...).

```
266 \def\EmbedFile@add#1#2{%
267   \begingroup
268   \edef\key{\pdfescapehex{#1}}%
269   \ifx\EmbedFile@list\empty
270     \xdef\EmbedFile@list{\noexpand\do{\key}{#2}}%
271   \else
272     \def\do##1##2{%
273       \ifnum\pdfstrcmp{##1}{\key}>0 %
274         \edef\x{%
275           \toks@{%
276             \the\toks@%
277             \noexpand\do{\key}{#2}%
278             \noexpand\do{##1}{##2}%
279           }%
280         }%
281         \x
282         \def\do####1####2{%
283           \toks@\expandafter{\the\toks@\do{####1}{####2}}%
284         }%
285         \def\stop{%
286           \xdef\EmbedFile@list{\the\toks@}%
287         }%
288       \else
289         \toks@\expandafter{\the\toks@\do{##1}{##2}}%
290       \fi
291     }%
292     \def\stop{%
293       \xdef\EmbedFile@list{\the\toks@\noexpand\do{\key}{#2}}%
294     }%
295     \toks@{}%
296     \EmbedFile@list\stop
297   \fi
298   \endgroup
299 }

300 \newif\ifEmbedFile@finished
```

\embedfilefinish

```
301 \def\embedfilefinish{%
302   \ifEmbedFile@finished
303     \EmbedFile@PackageError{%
304       Too many invocations of \string\embedfilefinish
305     }{%
306       The list of embedded files is already written.%
307     }%
308   \else
```



```

309 \ifx\EmbedFile@list\empty
310 \else
311 \global\EmbedFile@finishedtrue
312 \begingroup
313 \def\do##1##2{%
314 <##1>##2%
315 }%
316 \immediate\pdfobj{%
317 <<%
318 /Names[\EmbedFile@list]%
319 >>%
320 }%
321 \pdfnames{%
322 /EmbeddedFiles \the\pdflastobj\space 0 R%
323 }%
324 \endgroup
325 \fi
326 \fi
327 }

328 \begingroup\expandafter\expandafter\expandafter\endgroup
329 \expandafter\ifx\csname AtEndDocument\endcsname\relax
330 \else
331 \AtEndDocument{\embedfilefinish}%
332 \fi

333 \EmbedFileRestoreCatcodes
334 </package>

```

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/
```

¹<http://ftp.ctan.org/tex-archive/>

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
```

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>	\rightarrow	<code>tex/latex/oberdiek/embedfile.sty</code>
<code>dtx-attach.sty</code>	\rightarrow	<code>tex/latex/oberdiek/dtx-attach.sty</code>
<code>embedfile.pdf</code>	\rightarrow	<code>doc/latex/oberdiek/embedfile.pdf</code>
<code>embedfile-example-plain.tex</code>	\rightarrow	<code>doc/latex/oberdiek/embedfile-example-plain.tex</code>
<code>embedfile.dtx</code>	\rightarrow	<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.

[2007/04/11 v1.1]

- Line ends sanitized.

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	195
\@undefined	181
\\	208
A		
\AtEndDocument	331
B		
\bye	34
C		
\catcode	49, 50, 51, 52, 53, 77, 78, 79, 80, 81, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119
\csname	54, 64, 82, 95, 121, 135, 137, 143, 160, 172, 174, 329
D		
\define@key	171
\do	270, 272, 277, 278, 282, 283, 289, 293, 313
E		
\embedfile	..	2, 15, 20, 24, 41, <u>194</u> , 200
\EmbedFile@@desc	..	217, 219, 241, 243
\EmbedFile@@filespec	..	215, 240, 251
\EmbedFile@add	250, <u>266</u>
\EmbedFile@convert	215, 219, <u>257</u>
\EmbedFile@DefineKey 170, 176, 177, 178, 179, 180
\EmbedFile@desc	216, 219
\EmbedFile@embedfile	195, <u>197</u>
\EmbedFile@file	177, 206, 208, 210, 228, 229, 230, 232
\EmbedFile@filespec	215
\EmbedFile@filesystem	236, 238
\EmbedFile@finishedtrue	311
\EmbedFile@list	265, 269, 270, 286, 293, 296, 309, 318
\EmbedFile@mimetype	223, 225
\EmbedFile@PackageError <u>120</u> , 151, 161, 199, 209, 303
\EmbedFile@RequirePackage <u>134</u> , 148, 169
\EmbedFile@stringmethod	258
\EmbedFile@temp	259, 260
\embedfilefinish	...	33, 200, <u>301</u> , 331
\EmbedFileRestoreCatcodes 98, 156, 166, 333

<code>\embedfilesetup</code>	10, <u>191</u>		
<code>\empty</code>	58, 129, 216, 217, 223, 236, 241, 265, 269, 309		
<code>\encsname</code>	137		
<code>\endcsname</code>	54, 64, 82, 95, 121, 135, 143, 160, 172, 174, 329		
<code>\endingput</code>	73, 157, 167		
<code>\errhelp</code>	123		
<code>\errmessage</code>	124		
G			
<code>\Gin@driver</code>	4		
I			
<code>\ifcase</code>	55		
<code>\ifEmbedFile@finished</code> .	198, 300, 302		
<code>\ifnum</code>	258, 273		
<code>\ifpdf</code>	149		
<code>\ifx</code>	56, 58, 64, 82, 90, 121, 135, 137, 143, 160, 181, 184, 208, 216, 223, 236, 241, 269, 309, 329		
<code>\immediate</code>	66, 84, 221, 233, 316		
<code>\input</code>	3, 5, 6, 138		
J			
<code>\jobname</code>	39, 44, 45		
K			
<code>\key</code>	268, 270, 273, 277, 293		
N			
<code>\NeedsTeXFormat</code>	37		
<code>\newif</code>	300		
O			
<code>\on@line</code>	129		
P			
<code>\PackageError</code>	130		
<code>\PackageInfo</code>	69		
<code>\pdfescapehex</code>	268		
<code>\pdfescapename</code>	225, 238		
<code>\pdfescapestring</code>	262		
<code>\pdffilemoddate</code>	228		
<code>\pdffilesize</code>	208, 229		
<code>\pdflastobj</code>	246, 252, 322		
<code>\pdfmdfivesum</code>	230		
<code>\pdfnames</code>	321		
<code>\pdfobj</code>	221, 233, 316		
<code>\pdfstrcmp</code>	258, 273		
<code>\pdfstringdef</code>	181, 184, 259		
<code>\ProvidesPackage</code>	38, 96		
R			
<code>\RequirePackage</code>	40, 144		
<code>\resetatcatcode</code>	7		
S			
<code>\setkeys</code>	192, 207		
<code>\space</code>	200, 246, 252, 322		
<code>\stop</code>	285, 292, 296		
T			
<code>\the</code>	99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 246, 252, 276, 283, 286, 289, 293, 322		
<code>\toks@</code>	275, 276, 283, 286, 289, 293, 295		
W			
<code>\write</code>	66, 84		
X			
<code>\x</code>	54, 56, 58, 65, 69, 71, 83, 88, 95, 274, 281		