

# The embedfile package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/09/09 v1.2

## Abstract

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

## Contents

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Introduction	2
1.1.1	Future development	2
1.2	User interface	2
1.3	Examples	3
1.3.1	plain-T <sub>E</sub> X	3
1.4	Package dtx-attach	4
<b>2</b>	<b>Implementation</b>	<b>4</b>
2.1	Reload check and package identification	4
2.2	Catcodes	5
2.3	Tools	5
2.4	Check for recent pdfTeX in PDF mode	6
2.5	Key value definitions	6
2.6	Embed the file	8
<b>3</b>	<b>Test</b>	<b>11</b>
3.1	Catcode checks for loading	11
3.2	Simple test	11
<b>4</b>	<b>Installation</b>	<b>12</b>
4.1	Download	12
4.2	Bundle installation	12
4.3	Package installation	12
4.4	Refresh file name databases	13
4.5	Some details for the interested	13
<b>5</b>	<b>References</b>	<b>13</b>
<b>6</b>	<b>History</b>	<b>14</b>
	[2006/08/16 v1.0]	14
	[2007/04/11 v1.1]	14
	[2007/09/09 v1.2]	14
<b>7</b>	<b>Index</b>	<b>14</b>

# 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<sup>A</sup>T<sub>E</sub>X and plain-T<sub>E</sub>X. See [subsection 1.3.1](#) that explains the use with plain-T<sub>E</sub>X by an example. In L<sup>A</sup>T<sub>E</sub>X 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`.

`\embedfilefinish`

The list of all embedded files must be added as data structure in the PDF file. In case of L<sup>A</sup>T<sub>E</sub>X this is automatically done. The package uses `\AtEndDocument`. Then the list of all files should be known. However, plain-T<sub>E</sub>X does not know about `\AtEndDocument`. Thus the user must call `\embedfilefinish` at the end of the document after the last file is embedded.

## 1.3 Examples

### 1.3.1 plain-T<sub>E</sub>X

The package can be used with plain-T<sub>E</sub>X. It can be used with or without 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-T<sub>E</sub>X 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 \end{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<sup>A</sup>T<sub>E</sub>X.

```

36 \begin{dtxattach}
37 \NeedsTeXFormat{LaTeX2e}
38 \ProvidesPackage{dtx-attach}
39 [2007/09/09 v1.2 Embed \string\jobname.dtx (HO)]%
40 \RequirePackage{embedfile}[2007/09/09]
41 \embedfile[%
42   stringmethod=escape,%
43   mimetype=plain/text,%
44   desc={LaTeX docstrip source archive for package '\jobname'}%
45 ]{\jobname.dtx}
46 \end{dtxattach}

```

## 2 Implementation

```

47 \begin{package}

```

### 2.1 Reload check and package identification

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```

48 \begin{group}
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   \end{group}
73   \expandafter\endinput
74 \fi
75 \end{group}

```

Package identification:

```

76 \begin{group}
77   \catcode40 12 % (

```

```

78 \catcode41 12 % )
79 \catcode44 12 % ,
80 \catcode45 12 % -
81 \catcode46 12 % .
82 \catcode47 12 % /
83 \catcode58 12 % :
84 \catcode64 11 % @
85 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
86 \def\x#1#2#3[#4]{\endgroup
87 \immediate\write-1{Package: #3 #4}%
88 \xdef#1{#4}%
89 }%
90 \else
91 \def\x#1#2[#3]{\endgroup
92 #2[{#3}]%
93 \ifx#1\relax
94 \xdef#1{#3}%
95 \fi
96 }%
97 \fi
98 \expandafter\x\csname ver@embedfile.sty\endcsname
99 \ProvidesPackage{embedfile}%
100 [2007/09/09 v1.2 embed files into PDF (H0)]

```

## 2.2 Catcodes

```

101 \expandafter\edef\csname EmbedFile@AtEnd\endcsname{%
102 \catcode64 \the\catcode64\relax
103 }
104 \catcode64 11 % @
105 \def\TMP@EnsureCode#1#2{%
106 \edef\EmbedFile@AtEnd{%
107 \EmbedFile@AtEnd
108 \catcode#1 \the\catcode#1\relax
109 }%
110 \catcode#1 #2\relax
111 }
112 \TMP@EnsureCode{39}{12}% '
113 \TMP@EnsureCode{40}{12}% (
114 \TMP@EnsureCode{41}{12}% )
115 \TMP@EnsureCode{44}{12}% ,
116 \TMP@EnsureCode{46}{12}% .
117 \TMP@EnsureCode{47}{12}% /
118 \TMP@EnsureCode{58}{12}% :
119 \TMP@EnsureCode{60}{12}% <
120 \TMP@EnsureCode{61}{12}% =
121 \TMP@EnsureCode{62}{12}% >
122 \TMP@EnsureCode{91}{12}% [
123 \TMP@EnsureCode{93}{12}% ]
124 \TMP@EnsureCode{96}{12}% '

```

## 2.3 Tools

\EmbedFile@RequirePackage

```

125 \begingroup\expandafter\expandafter\expandafter\endgroup
126 \expandafter\ifx\csname RequirePackage\endcsname\relax
127 \def\EmbedFile@RequirePackage#1[#2]{%
128 \input #1.sty\relax
129 }%
130 \else
131 \let\EmbedFile@RequirePackage\RequirePackage
132 \fi

```

\EmbedFile@PackageError

```

133 \EmbedFile@RequirePackage{infwarerr}[2007/09/09]%
134 \def\EmbedFile@PackageError{%
135   \@PackageError{embedfile}%
136 }

```

## 2.4 Check for recent pdfTeX in PDF mode

Load package ifpdf and check mode.

```

137 \EmbedFile@RequirePackage{ifpdf}[2007/09/09]
138 \ifpdf
139 \else
140   \EmbedFile@PackageError{%
141     Missing pdfTeX in PDF mode%
142   }{%
143     Currently other drivers are not supported. %
144     Package loading is aborted.%
145   }%
146   \EmbedFile@AtEnd
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   \EmbedFile@AtEnd
157   \expandafter\endinput
158 \fi

```

## 2.5 Key value definitions

```

159 \expandafter\ifx\csname define@key\endcsname\relax
160   \chardef\EmbedFile@plain=\z@
161   \def\EmbedFile@temp#1{%
162     \begingroup\expandafter\expandafter\expandafter\endgroup
163     \expandafter\ifx\csname#1\endcsname\relax
164       \chardef\EmbedFile@plain=\@ne
165     \fi
166   }%
167   \EmbedFile@temp{NeedsTeXFormat}%
168   \EmbedFile@temp{ProvidesPackage}%
169   \EmbedFile@temp{DeclareOption}%
170   \EmbedFile@temp{ExecuteOptions}%
171   \EmbedFile@temp{ProcessOptions}%
172   \ifnum\EmbedFile@plain=\@ne
173     \def\EmbedFile@temp#1{%
174       \expandafter\let\csname EmbedFile@Org#1\expandafter\endcsname
175         \csname#1\endcsname
176       \expandafter\def\csname#1\endcsname
177     }%
178     \EmbedFile@temp{NeedsTeXFormat}#1{}%
179     \EmbedFile@temp{ProvidesPackage}#1[#2]{}% hash-ok
180     \EmbedFile@temp{DeclareOption}#1{}%
181     \EmbedFile@temp{ExecuteOptions}#1{}%
182     \EmbedFile@temp{ProcessOptions}{}%

```

\KV@errx L<sup>A</sup>T<sub>E</sub>X's option processing is not available with plain-T<sub>E</sub>X. Thus we define the default error command \KV@errx here, also using package infwarerr's \@PackageError.

```

183 \def\KV@errx#1{%
184   \@PackageError{keyval}{#1}\@ehc
185 }%

```

Other macros from L<sup>A</sup>T<sub>E</sub>X's kernel that are used by package keyval.

\@ifnextchar

```

186 \expandafter\ifx\csname @ifnextchar\endcsname\relax
187 \def\@ifnextchar#1#2#3{%
188   \let\reserved@d=#1%
189   \def\reserved@a{#2}%
190   \def\reserved@b{#3}%
191   \futurelet\@let@token\@ifnch
192 }%
193 \def\@ifnch{%
194   \ifx\@let@token\@sptoken
195     \let\reserved@c\@xifnch
196   \else
197     \ifx\@let@token\reserved@d
198       \let\reserved@c\reserved@a
199     \else
200       \let\reserved@c\reserved@b
201     \fi
202   \fi
203   \reserved@c
204 }%
205 \begingroup
206 \def\:{\global\let\@sptoken= }%
207 \: % this makes \@sptoken a space token
208 \def\:{\@xifnch}%
209 \expandafter\gdef\:{ %
210   \futurelet\@let@token\@ifnch
211 }%
212 \endgroup
213 \fi

```

\@namedef

```

214 \expandafter\ifx\csname @namedef\endcsname\relax
215 \def\@namedef#1{%
216   \expandafter\def\csname#1\endcsname
217 }%
218 \fi

219 \fi
220 \EmbedFile@RequirePackage{keyval}[1999/03/16]%
221 \ifnum\EmbedFile@plain=\@ne
222 \def\EmbedFile@temp#1{%
223   \expandafter\let\csname#1\expandafter\endcsname
224     \csname EmbedFile@Org#1\endcsname
225 }%
226 \EmbedFile@temp{NeedsTeXFormat}%
227 \EmbedFile@temp{ProvidesPackage}%
228 \EmbedFile@temp{DeclareOption}%
229 \EmbedFile@temp{ExecuteOptions}%
230 \EmbedFile@temp{ProcessOptions}%
231 \fi
232 \fi

233 \def\EmbedFile@DefineKey#1#2{%
234   \define@key{EmbedFile}{#1}{%
235     \expandafter\def\csname EmbedFile@#1\endcsname{##1}%
236   }%
237   \expandafter\def\csname EmbedFile@#1\endcsname{#2}%
238 }

```

Subtype of the embedded file (optional).

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

File specification string.

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

File system (optional).

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

Description (optional).

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

Method for converting text to PDF strings.

```
243 \EmbedFile@DefineKey{stringmethod}{%
```

```
244   \ifx\pdfstringdef\@undefined
```

```
245     escape%
```

```
246   \else
```

```
247     \ifx\pdfstringdef\relax
```

```
248       escape%
```

```
249     \else
```

```
250       psd%
```

```
251     \fi
```

```
252   \fi
```

```
253 }
```

```
\embedfilessetup
```

```
254 \def\embedfilessetup{%
```

```
255   \setkeys{EmbedFile}%
```

```
256 }
```

## 2.6 Embed the file

```
\embedfile
```

```
257 \def\embedfile{%
```

```
258   \@ifnextchar[\EmbedFile@embedfile{\EmbedFile@embedfile[]}%
```

```
259 }
```

```
\EmbedFile@embedfile
```

```
260 \def\EmbedFile@embedfile[#1]#2{%
```

```
261   \ifEmbedFile@finished
```

```
262     \EmbedFile@PackageError{%
```

```
263       \string\embedfile\space after \string\embedfilefinish
```

```
264     }{%
```

```
265       The list of embedded files is already written.%
```

```
266     }%
```

```
267   \else
```

```
268     \beginingroup
```

```
269     \def\EmbedFile@file{#2}%
```

```
270     \setkeys{EmbedFile}{#1}%
```

```
271     \expandafter\ifx\expandafter\\\pdffilesize{\EmbedFile@file}\\\%
```

```
272     \EmbedFile@PackageError{%
```

```
273       File '\EmbedFile@file' not found%
```

```
274     }{%
```

```
275       The unknown file is not embedded.%
```

```
276     }%
```

```
277   \else
```

```
278     \EmbedFile@convert\EmbedFile@filespec\EmbedFile@@filespec
```

```
279     \ifx\EmbedFile@desc\empty
```

```
280       \let\EmbedFile@@desc\empty
```

```
281     \else
```

```
282       \EmbedFile@convert\EmbedFile@desc\EmbedFile@@desc
```

```
283     \fi
```

```
284     \immediate\pdfobj stream attr{%
```



```

285         /Type/EmbeddedFile%
286         \ifx\EmbedFile@mimetype\empty
287         \else
288         /Subtype/\pdfescapename{\EmbedFile@mimetype}%
289         \fi
290         /Params<<%
291         /ModDate(\pdffilemoddate{\EmbedFile@file})%
292         /Size \pdffilesize{\EmbedFile@file}%
293         /Checksum<\pdfmdfivesum file{\EmbedFile@file}>%
294         >>%
295     }file{\EmbedFile@file}\relax
296     \immediate\pdfobj{%
297     <<%
298     /Type/Filespec%
299     \ifx\EmbedFile@filesystem\empty
300     \else
301     /FS/\pdfescapename{\EmbedFile@filesystem}%
302     \fi
303     /F(\EmbedFile@@filespec)%
304     \ifx\EmbedFile@@desc\empty
305     \else
306     /Desc(\EmbedFile@@desc)%
307     \fi
308     /EF<<%
309     /F \the\pdflastobj\space 0 R%
310     >>%
311     >>%
312     }%
313     \EmbedFile@add{%
314     \EmbedFile@@filespec
315     }{\the\pdflastobj\space 0 R}%
316     \fi
317     \endgroup
318     \fi
319 }

```

\EmbedFile@convert

```

320 \def\EmbedFile@convert#1#2{%
321     \ifnum\pdfstrcmp{\EmbedFile@stringmethod}{psd}=0 %
322     \pdfstringdef\EmbedFile@temp{#1}%
323     \let#2\EmbedFile@temp
324     \else
325     \edef#2{\pdfescapestring{#1}}%
326     \fi
327 }

328 \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, ...).

```

329 \def\EmbedFile@add#1#2{%
330     \begingroup
331     \edef\key{\pdfescapehex{#1}}%
332     \ifx\EmbedFile@list\empty
333     \xdef\EmbedFile@list{\noexpand\do{\key}{#2}}%
334     \else
335     \def\do##1##2{%
336         \ifnum\pdfstrcmp{##1}{\key}>0 %
337         \edef\x{%
338             \toks@{%

```

```

339         \the\toks@%
340         \noexpand\do{\key}{#2}%
341         \noexpand\do{##1}{##2}%
342     }%
343 }%
344 \x
345 \def\do####1####2{%
346     \toks@\expandafter{\the\toks@\do{####1}{####2}}%
347 }%
348 \def\stop{%
349     \xdef\EmbedFile@list{\the\toks@}%
350 }%
351 \else
352     \toks@\expandafter{\the\toks@\do{##1}{##2}}%
353 \fi
354 }%
355 \def\stop{%
356     \xdef\EmbedFile@list{\the\toks@\noexpand\do{\key}{#2}}%
357 }%
358 \toks@{}%
359 \EmbedFile@list\stop
360 \fi
361 \endgroup
362 }

363 \newif\ifEmbedFile@finished

\embedfilefinish
364 \def\embedfilefinish{%
365     \ifEmbedFile@finished
366         \EmbedFile@PackageError{%
367             Too many invocations of \string\embedfilefinish
368         }{%
369             The list of embedded files is already written.%
370         }%
371     \else
372         \ifx\EmbedFile@list\empty
373         \else
374             \global\EmbedFile@finishedtrue
375             \begingroup
376                 \def\do##1##2{%
377                     <##1>##2%
378                 }%
379                 \immediate\pdfobj{%
380                     <<%
381                         /Names[\EmbedFile@list]%
382                     >>%
383                 }%
384                 \pdfnames{%
385                     /EmbeddedFiles \the\pdflastobj\space 0 R%
386                 }%
387             \endgroup
388         \fi
389     \fi
390 }

391 \begingroup\expandafter\expandafter\expandafter\endgroup
392 \expandafter\ifx\csname AtEndDocument\endcsname\relax
393 \else
394     \AtEndDocument{\embedfilefinish}%
395 \fi

396 \EmbedFile@AtEnd
397 \end{package}

```

## 3 Test

### 3.1 Catcode checks for loading

```
398 <*test1>
399 \catcode'\@=11 %
400 \def\RestoreCatcodes{}
401 \count@=0 %
402 \loop
403   \edef\RestoreCatcodes{%
404     \RestoreCatcodes
405     \catcode\the\count@=\the\catcode\count@\relax
406   }%
407 \ifnum\count@<255 %
408   \advance\count@\@ne
409 \repeat
410
411 \def\RangeCatcodeInvalid#1#2{%
412   \count@=#1\relax
413   \loop
414     \catcode\count@=15 %
415     \ifnum\count@<#2\relax
416       \advance\count@\@ne
417     \repeat
418 }
419 \def\Test{%
420   \RangeCatcodeInvalid{0}{47}%
421   \RangeCatcodeInvalid{58}{64}%
422   \RangeCatcodeInvalid{91}{96}%
423   \RangeCatcodeInvalid{123}{255}%
424   \catcode'\@=12 %
425   \catcode'\=0 %
426   \catcode'\{=1 %
427   \catcode'\}=2 %
428   \catcode'\#=6 %
429   \catcode'\[=12 %
430   \catcode'\]=12 %
431   \catcode'\%=14 %
432   \catcode'\ =10 %
433   \catcode13=5 %
434   \input embedfile.sty\relax
435   \RestoreCatcodes
436 }
437 \Test
438 \csname @@end\endcsname
439 \end
440 </test1>
```

### 3.2 Simple test

```
441 <*test2>
442 \input embedfile.sty\relax
443 \embedfile[%
444   stringmethod=escape,%
445   mimetype=plain/text,%
446   desc={LaTeX docstrip source archive for package 'embedfile'}%
447 ]{embedfile.dtx}
448 \nopagenumbers
449 Test (plain-TeX): {\tt embedfile.dtx} should be embedded.%
450 \embedfilefinish
451 \bye
452 </test2>
```

```

453 \test3}
454 \NeedsTeXFormat{LaTeX2e}
455 \let\SavedJobname\jobname
456 \def\jobname{embedfile}
457 \RequirePackage{dtx-attach}[2007/09/09]
458 \let\jobname\SavedJobname
459 \documentclass{minimal}
460 \begin{document}
461   Test (\LaTeX): \texttt{embedfile.dtx} should be embedded.%
462 \end{document}
463 \test3}

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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](#)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 4.2 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/

```

### 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain-T<sub>E</sub>X:

```
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>	→ <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-test1.tex</code>	→ <code>doc/latex/oberdiek/embedfile-test1.tex</code>
<code>embedfile-test2.tex</code>	→ <code>doc/latex/oberdiek/embedfile-test2.tex</code>
<code>embedfile-test3.tex</code>	→ <code>doc/latex/oberdiek/embedfile-test3.tex</code>
<code>embedfile.dtx</code>	→ <code>source/latex/oberdiek/embedfile.dtx</code>

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

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`.

## 4.4 Refresh file name databases

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

## 4.5 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  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), 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
```

## 5 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](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/>.

## 6 History

[2006/08/16 v1.0]

- First public version.

[2007/04/11 v1.1]

- Line ends sanitized.

[2007/09/09 v1.2]

- Fixes for plain-TeX, wrapper for package keyval added.
- Catcode section rewritten.

## 7 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	
\#	428
\%	431
\:	206, 207, 208, 209
\@	399, 424
\@PackageError	135, 184
\@ehc	184
\@ifnch	191, 193, 210
\@ifnextchar	186, 258
\@let@token	191, 194, 197, 210
\@namedef	214
\@ne	164, 172, 221, 408, 416
\@sptoken	194, 206, 207
\@undefined	244
\@xifnch	195, 208
\[	429
\]	271, 425
\{	426
\}	427
\]	430
\_	432
A	
\advance	408, 416
\AtEndDocument	394
B	
\begin	460
\bye	34, 451
C	
\catcode	49, 50, 51, 52, 53, 77, 78, 79, 80, 81, 82, 83, 84, 102, 104, 108, 110, 399, 405, 414, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433
\chardef	160, 164
\count@	401, 405, 407, 408, 412, 414, 415, 416
\csname	54, 64, 85, 98, 101, 126, 150, 159, 163, 174, 175, 176, 186, 214, 216, 223, 224, 235, 237, 392, 438
D	
\define@key	234
\do	333, 335, 340, 341, 345, 346, 352, 356, 376
\documentclass	459
E	
\embedfile	2, 15, 20, 24, 41, 257, 263, 443
\EmbedFile@desc	280, 282, 304, 306
\EmbedFile@filespec	278, 303, 314
\EmbedFile@add	313, 329
\EmbedFile@AtEnd	106, 107, 146, 156, 396
\EmbedFile@convert	278, 282, 320
\EmbedFile@DefineKey	233, 239, 240, 241, 242, 243
\EmbedFile@desc	279, 282
\EmbedFile@embedfile	258, 260
\EmbedFile@file	240, 269, 271, 273, 291, 292, 293, 295
\EmbedFile@filespec	278
\EmbedFile@filesystem	299, 301
\EmbedFile@finishedtrue	374
\EmbedFile@list	328, 332, 333, 349, 356, 359, 372, 381
\EmbedFile@mimetype	286, 288
\EmbedFile@PackageError	133, 140, 151, 262, 272, 366
\EmbedFile@plain	160, 164, 172, 221
\EmbedFile@RequirePackage	125, 133, 137, 220
\EmbedFile@stringmethod	321
\EmbedFile@temp	161, 167, 168, 169, 170, 171, 173,

178, 179, 180, 181, 182, 222, 226, 227, 228, 229, 230, 322, 323	\pdfescapename . . . . . 288, 301
\embedfilefinish . . . . .	\pdfescapestring . . . . . 325
. . . . . 3, 33, 263, 364, 394, 450	\pdffilemoddate . . . . . 291
\embedfilesetup . . . . . 10, 254	\pdffilesize . . . . . 271, 292
\empty . . . . . 58, 279,	\pdflastobj . . . . . 309, 315, 385
280, 286, 299, 304, 328, 332, 372	\pdfmdfivesum . . . . . 293
\end . . . . . 439, 462	\pdfnames . . . . . 384
\endcsname . . . . . 54,	\pdfobj . . . . . 284, 296, 379
64, 85, 98, 101, 126, 150, 159,	\pdfstrcmp . . . . . 321, 336
163, 174, 175, 176, 186, 214,	\pdfstringdef . . . . . 244, 247, 322
216, 223, 224, 235, 237, 392, 438	\ProvidesPackage . . . . . 38, 99
\endinput . . . . . 73, 147, 157	
	<b>R</b>
<b>F</b>	\RangeCatcodeInvalid . . . . .
\futurelet . . . . . 191, 210	. . . . . 411, 420, 421, 422, 423
	\repeat . . . . . 409, 417
<b>G</b>	\RequirePackage . . . . . 40, 131, 457
\gdef . . . . . 209	\reserved@a . . . . . 189, 198
\Gin@driver . . . . . 4	\reserved@b . . . . . 190, 200
	\reserved@c . . . . . 195, 198, 200, 203
<b>I</b>	\reserved@d . . . . . 188, 197
\ifcase . . . . . 55	\resetatcatcode . . . . . 7
\ifEmbedFile@finished . . . . . 261, 363, 365	\RestoreCatcodes . . . . . 400, 403, 404, 435
\ifnum . . . . . 172, 221, 321, 336, 407, 415	
\ifpdf . . . . . 138	<b>S</b>
\ifx . . . . . 56, 58, 64,	\SavedJobname . . . . . 455, 458
85, 93, 126, 150, 159, 163, 186,	\setkeys . . . . . 255, 270
194, 197, 214, 244, 247, 271,	\space . . . . . 263, 309, 315, 385
279, 286, 299, 304, 332, 372, 392	\stop . . . . . 348, 355, 359
\immediate . . . . . 66, 87, 284, 296, 379	
\input . . . . . 3, 5, 6, 128, 434, 442	<b>T</b>
	\Test . . . . . 419, 437
<b>J</b>	\TeX . . . . . 449
\jobname . . . . . 39, 44, 45, 455, 456, 458	\texttt . . . . . 461
	\the . . . . . 102, 108, 309, 315,
<b>K</b>	339, 346, 349, 352, 356, 385, 405
\key . . . . . 331, 333, 336, 340, 356	\TMP@EnsureCode . . . . . 105,
\KV@errx . . . . . 183	112, 113, 114, 115, 116, 117,
	118, 119, 120, 121, 122, 123, 124
<b>L</b>	\toks@ 338, 339, 346, 349, 352, 356, 358
\LaTeX . . . . . 461	\tt . . . . . 449
\loop . . . . . 402, 413	
	<b>W</b>
<b>N</b>	\write . . . . . 66, 87
\NeedsTeXFormat . . . . . 37, 454	
\newif . . . . . 363	<b>X</b>
\nopagenumbers . . . . . 448	\x . . . . . 54, 56,
	58, 65, 69, 71, 86, 91, 98, 337, 344
<b>P</b>	
\PackageInfo . . . . . 69	<b>Z</b>
\pdfescapehex . . . . . 331	\z@ . . . . . 160