The **embedfile** package

Heiko Oberdiek
<*heiko.oberdiek at googlemail.com*>

2011/04/13 v2.6

Abstract

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

Contents

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 meantioned 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 LaTeX and plain TeX. See subsubsection 1.5.1 that explains the use with plain TeX by an example. In LaTeX the package is loaded as usually. There are no options.

```
\usepackage{embedfile}
```

2

> `\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 (8-bit, path separators, ...), look into the PDF specification ([3]). There are rules how these file names must be written/encoded. Avoid 8-bit characters and other special characters, the behaviour is currently undefined. Use option ucfilespec for more funny file names. The string method, see below, is escape since version 2.4.

This name is also used as entry in a name tree (see PDF specification: `/EmbeddedFiles`). Therefore the value for filespec must be unique among all embedded files. Also key initialfiles refers to this name, if the file name and the value of filespec are different.

ucfilespec Since PDF 1.7 the file name may be provided in Unicode. The conversion of the option value into a PDF string is controlled by option string-method.

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 1.5.1 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 ucfilespec and desc into a PDF string (before version 2.4: filespec and desc). 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`.

⟨**key**⟩**.value** Sets the value of a collection item property, see section 1.3.

⟨**key**⟩**.prefix** Sets the prefix of a collection item property, see section 1.3.

id The value must be an unique name. Macros `\embedfileifobjectexists` and `\embedfilegetobject` are using this name later.

> `\embedfilefinish`

The list of all embedded files must be added as data structure in the PDF file. In case of LaTeX this is automatically done. The package uses `\AtEndDocument`. Then the list of all files should be known. However, plain TeX does not know about `\AtEndDocument`. Thus the user must call `\embedfilefinish` at the end of the document after the last file is embedded.

> `\embedfilesetup {⟨options⟩}`

Options for `\embedfile` and collection support can be set in `\embedfilesetup`.

1.3 Collection support (PDF 1.7)

Since PDF 1.7 the embedded files can form a *collection* (sometimes referred as *package*), the main document is called *cover sheet*. See PDF specification 8.2.4 "Collections" and 3.10.5 "Collection items" [3].

Usually Acrobat Reader 7 or 8 shows the embedded files in a table at the bottom with the following columns:

Name	Description	Modified	Size
...	...	...	...

Acrobat Reader 10 shows the embeded files in the left panel and adds a new column for the compressed size.

If the files form a collection, then they are displayed in a table left or top (depending on option view, see \embedfilesetup).

Collection support is enabled automatically, if it is used.

`\embedfilesetup {⟨options⟩}`

The following options are supported in addition to options for \embedfile:

view If the PDF file contains a collection, then Acrobat Reader 8 shows a line at the top below the menu bar and the toolbar. It shows the current selected file, icons for changing the view mode, an options menu. The initial mode how the collection is presented is set by this option view. The following modes/values are supported, the default is `details`:

   `details` The full collection table is displayed at the top below the collection bar.

   `tile` The files of the collection are shown in tile mode on the left.

   `hidden` The collection table is not shown.

initialfile Selects the file that is initially presented. Especially useful for an embedded PDF file that is then shown instead of the cover document. There must be an \embedfile command somewhere whose value for key filespec is used here. The \embedfile command can drop option filespec if the file name is not different.

`\embedfilefield {⟨key⟩} {⟨options⟩}`

Macro \embedfilefield defines a column/field in the collection table. The name of the field is ⟨key⟩.

type sets the type of the field. The supported values are:

   `text` A text field. Its value is set in \embedfile by option ⟨*key*⟩.value.

   `date` A date field. Its value is set in \embedfile by option ⟨*key*⟩.value. A special format is required, see "3.8.3 Dates" [3].

   `number` A field with an integer or float number. Its value is set in \embedfile by option ⟨*key*⟩.value.

   `file` The file name of the embedded file.

   `desc` The description text of the embedded file. It is set in \embedfile by option desc.

   `moddate` The modification date of the embedded file.

   `size` The size of the embedded file.

   All types allow the use of a prefix that is disregarded by sorting. The prefix for this field is set in \embedfile by option ⟨*key*⟩.prefix.

title sets the column title.

visible controls whether the column is presented:

> **true** shows the column.
>
> **false** hides the column.
>
> Default: **true**

edit Allows the editing of field values. Does not seem to have an effect for Acrobat Reader.

> **true** enables the feature, if available (depends on the PDF viewer).
>
> **false** disables the feature.
>
> Default: **false**

The order of \embedfilefield statements defines the order of the columns.

> `\embedfilesort {⟨key-sort-list⟩}`

The sort order of the embedded files are controlled by macro \embedfilesort. ⟨*key-sort-list*⟩ defines the sort order. The key is a field name defined by \embedfilefield. Its value is either **ascending** or **descending**. The default is **ascending**.

1.4 Export of object references

Caution: This feature is still experimental. It may be even removed in future versions. Therefore feedback would be nice, if someone has a useful application for this feature.

Object numbers are saved, if id is given in \embedfile. The following objects are supported:

- `EmbeddedFile`
- `Filespec`

> `\embedfileifobjectexists {⟨id⟩} {⟨type⟩} {⟨then⟩} {⟨else⟩}`

Macro \embedfileifobjectexists tests whether object of ⟨*type*⟩ is available for the embedded file identified by ⟨*id*⟩.

> `\embedfilegetobject {⟨id⟩} {⟨type⟩}`

Macro \embedfilegetobject expands to the full object reference object of ⟨*type*⟩ for the embedded file identified by ⟨*id*⟩.

1.4.1 Example

```
\embedfile[id={foo}]{foo.pdf}
\embedfileifobjectexists{foo}{Filespec}{%
  \typeout{%
    FileSpec object for 'foo': %
    \embedfilegetobject{foo}{Filespec}%
  }%
}{%
  \typeout{No Filespec object for 'foo'}%
}
```

1.5 Examples

1.5.1 plain TeX

The package can be used with plain TeX. 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 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 %<<END
 3 % Load packages
 4 \input miniltx
 5 % \def\Gin@driver{pdftex.def}
 6 % \input graphicx.sty
 7 \input embedfile.sty
 8 \resetatcatcode
 9
10 % default setting
11 \embedfilesetup{
12   mimetype=text/plain
13 }
14
15 % Embed files
16 \embedfile[
17   filespec=example.tex,
18   desc={Source code (plain-TeX) of this example}
19 ]{embedfile-example-plain.tex}
20
21 \embedfile[
22   desc={Source of package 'embedfile'}
23 ]{embedfile.dtx}
24
25 \embedfile[
26   mimetype=application/pdf,
27   desc={Documentation of package 'embedfile'}
28 ]{embedfile.pdf}
29
30 % Some text
31 This example document contains three embedded files.
32
33 % End of document
34 \embedfilefinish % don't forget
35 \bye
36 %END
37 ⟨/exampleplain⟩
```

1.5.2 Collection example

```
38 ⟨*examplecollection⟩
39 %<<END
40 \NeedsTeXFormat{LaTeX2e}
41 \documentclass{article}
42 \usepackage[bookmarks=false]{hyperref}
43 % provides \pdfstringdef that is then used by 'title' and
44 % other keys.
45 \usepackage{embedfile}[2011/04/13]
46 \embedfilesetup{
47   view=details,
48   initialfile=embedfile.pdf
49 }
50 \embedfilefield{file}{
```

```
51    type=file,
52    title={File name}
53 }
54 \embedfilefield{description}{
55    type=desc,
56    title={Description}
57 }
58 \embedfilefield{date}{
59    type=moddate,
60    title={Date}
61 }
62 \embedfilefield{size}{
63    type=size,
64    title={Size}
65 }
66 \embedfilefield{type}{
67    type=text,
68    title={Type},
69    visible=false
70 }
71 \embedfilesort{
72    type,
73    date=descending
74 }
75 \begin{document}
76 An example for embedded files as collection.
77 You need Acrobat Reader 8 or higher.
78
79 \embedfile[
80    desc={Source file of package 'embedfile'},
81    description.prefix={Package: },
82    type.value={DTX}
83 ]{embedfile.dtx}
84
85 \embedfile[
86    desc={Documentation of package 'embedfile'},
87    description.prefix={Package: },
88    type.value={PDF}
89 ]{embedfile.pdf}
90
91 \embedfile[
92    desc={The source for this example},
93    description.prefix={Example: },
94    type.value={TEX}
95 ]{\jobname.tex}
96
97 \end{document}
98 %END
99 ⟨/examplecollection⟩
```

1.6 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 LaTeX.

```
100 ⟨*dtxattach⟩
101 \NeedsTeXFormat{LaTeX2e}
102 \ProvidesPackage{dtx-attach}
103    [2011/04/13 v2.6 Embed \string\jobname.dtx (HO)]%
104 \RequirePackage{embedfile}[2011/04/13]
105 \embedfile[%
106    stringmethod=escape,%
107    mimetype=plain/text,%
```

```
108   desc={LaTeX docstrip source archive for package '\jobname'}%
109 ]{\jobname.dtx}
110 ⟨/dtxattach⟩
```

2   Implementation

```
111 ⟨*package⟩
```

2.1   Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
112 \begingroup\catcode61\catcode48\catcode32=10\relax%
113   \catcode13=5 % ^^M
114   \endlinechar=13 %
115   \catcode35=6 % #
116   \catcode39=12 % '
117   \catcode44=12 % ,
118   \catcode45=12 % -
119   \catcode46=12 % .
120   \catcode58=12 % :
121   \catcode64=11 % @
122   \catcode123=1 % {
123   \catcode125=2 % }
124   \expandafter\let\expandafter\x\csname ver@embedfile.sty\endcsname
125   \ifx\x\relax % plain-TeX, first loading
126   \else
127     \def\empty{}%
128     \ifx\x\empty % LaTeX, first loading,
129       % variable is initialized, but \ProvidesPackage not yet seen
130     \else
131       \expandafter\ifx\csname PackageInfo\endcsname\relax
132         \def\x#1#2{%
133           \immediate\write-1{Package #1 Info: #2.}%
134         }%
135       \else
136         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
137       \fi
138       \x{embedfile}{The package is already loaded}%
139       \aftergroup\endinput
140     \fi
141   \fi
142 \endgroup%
```

Package identification:

```
143 \begingroup\catcode61\catcode48\catcode32=10\relax%
144   \catcode13=5 % ^^M
145   \endlinechar=13 %
146   \catcode35=6 % #
147   \catcode39=12 % '
148   \catcode40=12 % (
149   \catcode41=12 % )
150   \catcode44=12 % ,
151   \catcode45=12 % -
152   \catcode46=12 % .
153   \catcode47=12 % /
154   \catcode58=12 % :
155   \catcode64=11 % @
156   \catcode91=12 % [
157   \catcode93=12 % ]
158   \catcode123=1 % {
159   \catcode125=2 % }
160   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
161     \def\x#1#2#3[#4]{\endgroup
```

```
162       \immediate\write-1{Package: #3 #4}%
163       \xdef#1{#4}%
164     }%
165   \else
166     \def\x#1#2[#3]{\endgroup
167       #2[{#3}]%
168       \ifx#1\@undefined
169         \xdef#1{#3}%
170       \fi
171       \ifx#1\relax
172         \xdef#1{#3}%
173       \fi
174     }%
175   \fi
176 \expandafter\x\csname ver@embedfile.sty\endcsname
177 \ProvidesPackage{embedfile}%
178   [2011/04/13 v2.6 embed files into PDF (HO)]%
```

2.2   Catcodes

```
179 \begingroup\catcode61\catcode48\catcode32=10\relax%
180   \catcode13=5 % ^^M
181   \endlinechar=13 %
182   \catcode123=1 % {
183   \catcode125=2 % }
184   \catcode64=11 % @
185   \def\x{\endgroup
186     \expandafter\edef\csname EmFi@AtEnd\endcsname{%
187       \endlinechar=\the\endlinechar\relax
188       \catcode13=\the\catcode13\relax
189       \catcode32=\the\catcode32\relax
190       \catcode35=\the\catcode35\relax
191       \catcode61=\the\catcode61\relax
192       \catcode64=\the\catcode64\relax
193       \catcode123=\the\catcode123\relax
194       \catcode125=\the\catcode125\relax
195     }%
196   }%
197 \x\catcode61\catcode48\catcode32=10\relax%
198 \catcode13=5 % ^^M
199 \endlinechar=13 %
200 \catcode35=6 % #
201 \catcode64=11 % @
202 \catcode123=1 % {
203 \catcode125=2 % }
204 \def\TMP@EnsureCode#1#2{%
205   \edef\EmFi@AtEnd{%
206     \EmFi@AtEnd
207     \catcode#1=\the\catcode#1\relax
208   }%
209   \catcode#1=#2\relax
210 }
211 \TMP@EnsureCode{39}{12}% '
212 \TMP@EnsureCode{40}{12}% (
213 \TMP@EnsureCode{41}{12}% )
214 \TMP@EnsureCode{44}{12}% ,
215 \TMP@EnsureCode{46}{12}% .
216 \TMP@EnsureCode{47}{12}% /
217 \TMP@EnsureCode{58}{12}% :
218 \TMP@EnsureCode{60}{12}% <
219 \TMP@EnsureCode{62}{12}% >
220 \TMP@EnsureCode{91}{12}% [
```

```
221 \TMP@EnsureCode{93}{12}% ]
222 \TMP@EnsureCode{96}{12}% '
223 \edef\EmFi@AtEnd{\EmFi@AtEnd\noexpand\endinput}
```

2.3 Tools

\EmFi@RequirePackage

```
224 \begingroup\expandafter\expandafter\expandafter\endgroup
225 \expandafter\ifx\csname RequirePackage\endcsname\relax
226   \def\EmFi@RequirePackage#1[#2]{%
227     \input #1.sty\relax
228   }%
229 \else
230   \let\EmFi@RequirePackage\RequirePackage
231 \fi
```

\EmFi@Error

```
232 \EmFi@RequirePackage{infwarerr}[2007/09/09]%
233 \def\EmFi@Error{%
234   \@PackageError{embedfile}%
235 }
```

2.4 Check for recent pdfTeX in PDF mode

Load package ifpdf and check mode.

```
236 \EmFi@RequirePackage{ifpdf}[2007/09/09]
237 \ifpdf
238 \else
239   \EmFi@Error{%
240     Missing pdfTeX in PDF mode%
241   }{%
242     Currently other drivers are not supported. %
243     Package loading is aborted.%
244   }%
245   \expandafter\EmFi@AtEnd
246 \fi%
247 \EmFi@RequirePackage{pdftexcmds}[2007/11/11]
248 \EmFi@RequirePackage{ltxcmds}[2010/03/01]
249 \EmFi@RequirePackage{kvsetkeys}[2010/03/01]
250 \EmFi@RequirePackage{kvdefinekeys}[2010/03/01]
```

Check version.

```
251 \begingroup\expandafter\expandafter\expandafter\endgroup
252 \expandafter\ifx\csname pdf@filesize\endcsname\relax
253   \EmFi@Error{%
254     Unsupported pdfTeX version%
255   }{%
256     At least version 1.30 is necessary. Package loading is aborted.%
257   }%
258   \expandafter\EmFi@AtEnd
259 \fi%
```

2.5 Strings

Minimal version of package pdfescape is 2007/08/27 v1.5 because of \EdefSanitize.

```
260 \EmFi@RequirePackage{pdfescape}[2007/11/11]
261 \def\EmFi@temp#1{%
262   \expandafter\EdefSanitize\csname EmFi@S@#1\endcsname{#1}%
263 }
```

```

\EmFi@details

```
264 \EmFi@temp{details}%
```

\EmFi@tile

```
265 \EmFi@temp{tile}%
```

\EmFi@hidden

```
266 \EmFi@temp{hidden}%
```

\EmFi@S@text

```
267 \EmFi@temp{text}
```

\EmFi@S@date

```
268 \EmFi@temp{date}
```

\EmFi@S@number

```
269 \EmFi@temp{number}
```

\EmFi@S@file

```
270 \EmFi@temp{file}
```

\EmFi@S@desc

```
271 \EmFi@temp{desc}
```

\EmFi@S@moddate

```
272 \EmFi@temp{moddate}
```

\EmFi@S@creationdate

```
273 \EmFi@temp{creationdate}
```

\EmFi@S@size

```
274 \EmFi@temp{size}
```

\EmFi@S@ascending

```
275 \EmFi@temp{ascending}
```

\EmFi@S@descending

```
276 \EmFi@temp{descending}
```

\EmFi@S@true

```
277 \EmFi@temp{true}
```

\EmFi@S@false

```
278 \EmFi@temp{false}
```

2.6 Switches

\ifEmFi@collection

```
279 \ltx@newif\ifEmFi@collection
```

\ifEmFi@sort

```
280 \ltx@newif\ifEmFi@sort
```

\ifEmFi@visible

```
281 \ltx@newif\ifEmFi@visible
```

\ifEmFi@edit

```
282 \ltx@newif\ifEmFi@edit
```

11

`\ifEmFi@item`

```
283 \ltx@newif\ifEmFi@item
```

`\ifEmFi@finished`

```
284 \ltx@newif\ifEmFi@finished
```

`\ifEmFi@id`

```
285 \ltx@newif\ifEmFi@id
```

2.7 Key value definitions

`\EmFi@GlobalKey`

```
286 \def\EmFi@GlobalKey#1#2{%
287   \global\expandafter\let\csname KV@#1@#2\expandafter\endcsname
288                         \csname KV@#1@#2\endcsname
289 }
```

`\EmFi@GlobalDefaultKey`

```
290 \def\EmFi@GlobalDefaultKey#1#2{%
291   \EmFi@GlobalKey{#1}{#2}%
292   \global\expandafter\let
293       \csname KV@#1@#2@default\expandafter\endcsname
294       \csname KV@#1@#2@default\endcsname
295 }
```

`\EmFi@DefineKey`

```
296 \def\EmFi@DefineKey#1#2{%
297   \kv@define@key{EmFi}{#1}{%
298     \expandafter\def\csname EmFi@#1\endcsname{##1}%
299   }%
300   \expandafter\def\csname EmFi@#1\endcsname{#2}%
301 }
```

Subtype of the embedded file (optional).

```
302 \EmFi@DefineKey{mimetype}{}
```

File specification string.

```
303 \EmFi@DefineKey{filespec}{\EmFi@file}
```

File specification string in Unicode.

```
304 \EmFi@DefineKey{ucfilespec}{}
```

File system (optional).

```
305 \EmFi@DefineKey{filesystem}{}
```

Description (optional).

```
306 \EmFi@DefineKey{desc}{}
```

Method for converting text to PDF strings.

```
307 \EmFi@DefineKey{stringmethod}{%
308   \ifx\pdfstringdef\@undefined
309     escape%
310   \else
311     \ifx\pdfstringdef\relax
312       escape%
313     \else
314       psd%
315     \fi
316   \fi
317 }
```

Option id as key for object numbers.

```
318 \kv@define@key{EmFi}{id}{%
319   \def\EmFi@id{#1}%
320   \EmFi@idtrue
321 }
```

\EmFi@defobj

```
322 \def\EmFi@defobj#1{%
323   \ifEmFi@id
324     \expandafter\xdef\csname EmFi@#1@\EmFi@id\endcsname{%
325       \the\pdflastobj\ltx@space 0 R%
326     }%
327   \fi
328 }
```

\embedfileifobjectexists

```
329 \def\embedfileifobjectexists#1#2{%
330   \expandafter\ifx\csname EmFi@#2@#1\endcsname\relax
331     \expandafter\ltx@secondoftwo
332   \else
333     \expandafter\ltx@firstoftwo
334   \fi
335 }
```

\embedfilegetobject

```
336 \def\embedfilegetobject#1#2{%
337   \embedfileifobjectexists{#1}{#2}{%
338     \csname EmFi@#2@#1\endcsname
339   }{%
340     0 0 R%
341   }%
342 }
```

Initial view of the collection.

```
343 \kv@define@key{EmFi}{view}[]{%
344   \EdefSanitize\EmFi@temp{#1}%
345   \def\EmFi@next{%
346     \global\EmFi@collectiontrue
347   }%
348   \ifx\EmFi@temp\ltx@empty
349     \let\EmFi@view\EmFi@S@details
350   \else\ifx\EmFi@temp\EmFi@S@details
351     \let\EmFi@view\EmFi@S@details
352   \else\ifx\EmFi@temp\EmFi@S@tile
353     \let\EmFi@view\EmFi@S@tile
354   \else\ifx\EmFi@temp\EmFi@S@hidden
355     \let\EmFi@view\EmFi@S@hidden
356   \else
357     \let\EmFi@next\relax
358     \EmFi@Error{%
359       Unknown value `\EmFi@temp' for key `view'.\MessageBreak
360       Supported values: `details', `tile', `hidden'.%
361     }\@ehc
362   \fi\fi\fi\fi
363   \EmFi@next
364 }
```

```
365 \EmFi@DefineKey{initialfile}{}
```

\embedfilesetup

```
366 \def\embedfilesetup{%
367   \ifEmFi@finished
```

```
368     \def\EmFi@next##1{}%
369     \EmFi@Error{%
370       \string\embedfilefield\ltx@space after \string\embedfilefinish
371     }{%
372       The list of embedded files is already written.%
373     }%
374   \else
375     \def\EmFi@next{%
376       \kvsetkeys{EmFi}%
377     }%
378   \fi
379   \EmFi@next
380 }
```

```
381 \def\EmFi@schema{}
```

```
382 \gdef\EmFi@order{0}
```

```
383 \let\EmFi@@order\relax
```

```
384 \def\EmFi@fieldlist{}
```

```
385 \def\EmFi@sortcase{0}%
```

```
386 \def\embedfilefield#1#2{%
387   \ifEmFi@finished
388     \EmFi@Error{%
389       \string\embedfilefield\ltx@space after \string\embedfilefinish
390     }{%
391       The list of embedded files is already written.%
392     }%
393   \else
394     \global\EmFi@collectiontrue
395     \EdefSanitize\EmFi@key{#1}%
396     \expandafter\ifx\csname KV@EmFi@\EmFi@key.prefix\endcsname\relax
397       \begingroup
398         \count@=\EmFi@order
399         \advance\count@ 1 %
400         \xdef\EmFi@order{\the\count@}%
401         \let\EmFi@title\EmFi@key
402         \let\EmFi@type\EmFi@S@text
403         \EmFi@visibletrue
404         \EmFi@editfalse
405         \kvsetkeys{EmFiFi}{#2}%
406         \EmFi@convert\EmFi@title\EmFi@title
407         \xdef\EmFi@schema{%
408           \EmFi@schema
409           /\pdf@escapename{\EmFi@key}<<%
410             /Subtype/%
411             \ifx\EmFi@type\EmFi@S@date D%
412             \else\ifx\EmFi@type\EmFi@S@number N%
413             \else\ifx\EmFi@type\EmFi@S@file F%
414             \else\ifx\EmFi@type\EmFi@S@desc Desc%
415             \else\ifx\EmFi@type\EmFi@S@moddate ModDate%
416             \else\ifx\EmFi@type\EmFi@S@creationdate CreationDate%
```

```
417        \else\ifx\EmFi@type\EmFi@S@size Size%
418        \else S%
419        \fi\fi\fi\fi\fi\fi\fi
420        /N(\EmFi@title)%
421        \EmFi@@order{\EmFi@order}%
422        \ifEmFi@visible
423        \else
424          /V false%
425        \fi
426        \ifEmFi@edit
427          /E true%
428        \fi
429      >>%
430    }%
431    \let\do\relax
432    \xdef\EmFi@fieldlist{%
433      \EmFi@fieldlist
434      \do{\EmFi@key}%
435    }%
436    \ifx\EmFi@type\EmFi@S@text
437      \kv@define@key{EmFi}{\EmFi@key.value}{%
438        \EmFi@itemtrue
439        \def\EmFi@temp{##1}%
440        \EmFi@convert\EmFi@temp\EmFi@temp
441        \expandafter\def\csname EmFi@V@#1%
442        \expandafter\endcsname\expandafter{%
443          \expandafter(\EmFi@temp)%
444        }%
445      }%
446      \EmFi@GlobalKey{EmFi}{\EmFi@key.value}%
447    \else\ifx\EmFi@type\EmFi@S@date
448      \kv@define@key{EmFi}{\EmFi@key.value}{%
449        \EmFi@itemtrue
450        \def\EmFi@temp{##1}%
451        \EmFi@convert\EmFi@temp\EmFi@temp
452        \expandafter\def\csname EmFi@V@#1%
453        \expandafter\endcsname\expandafter{%
454          \expandafter(\EmFi@temp)%
455        }%
456      }%
457      \EmFi@GlobalKey{EmFi}{\EmFi@key.value}%
458    \else\ifx\EmFi@type\EmFi@S@number
459      \kv@define@key{EmFi}{\EmFi@key.value}{%
460        \EmFi@itemtrue
461        \expandafter\EdefSanitize\csname EmFi@V@#1\endcsname{ ##1}%
462      }%
463      \EmFi@GlobalKey{EmFi}{\EmFi@key.value}%
464    \fi\fi\fi
465    \kv@define@key{EmFi}{\EmFi@key.prefix}{%
466      \EmFi@itemtrue
467      \expandafter\def\csname EmFi@P@#1\endcsname{##1}%
468    }%
469    \EmFi@GlobalKey{EmFi}{\EmFi@key.prefix}%
470    \kv@define@key{EmFiSo}{\EmFi@key}[ascending]{%
471      \EdefSanitize\EmFi@temp{##1}%
472      \ifx\EmFi@temp\EmFi@S@ascending
473        \def\EmFi@temp{true}%
474      \else\ifx\EmFi@temp\EmFi@S@descending
475        \def\EmFi@temp{false}%
476      \else
477        \def\EmFi@temp{}%
478        \EmFi@Error{%
```

```
479            Unknown sort order '\EmFi@temp'.\MessageBreak
480            Supported values: '\EmFi@S@ascending', %
481            '\EmFi@S@descending
482          }\@ehc
483        \fi\fi
484        \ifx\EmFi@temp\ltx@empty
485        \else
486          \xdef\EmFi@sortkeys{%
487            \EmFi@sortkeys
488            /\pdf@escapename{#1}%
489          }%
490          \ifx\EmFi@sortorders\ltx@empty
491            \global\let\EmFi@sortorders\EmFi@temp
492            \gdef\EmFi@sortcase{1}%
493          \else
494            \xdef\EmFi@sortorders{%
495              \EmFi@sortorders
496              \ltx@space
497              \EmFi@temp
498            }%
499            \xdef\EmFi@sortcase{2}%
500          \fi
501        \fi
502      }%
503      \EmFi@GlobalDefaultKey{EmFiSo}\EmFi@key
504    \endgroup
505  \else
506    \EmFi@Error{%
507      Field '\EmFi@key' is already defined%
508    }\@ehc
509  \fi
510  \fi
511 }

512 \kv@define@key{EmFiFi}{type}{%
513  \EdefSanitize\EmFi@temp{#1}%
514  \ifx\EmFi@temp\EmFi@S@text
515    \let\EmFi@type\EmFi@temp
516  \else\ifx\EmFi@temp\EmFi@S@date
517    \let\EmFi@type\EmFi@temp
518  \else\ifx\EmFi@temp\EmFi@S@number
519    \let\EmFi@type\EmFi@temp
520  \else\ifx\EmFi@temp\EmFi@S@file
521    \let\EmFi@type\EmFi@temp
522  \else\ifx\EmFi@temp\EmFi@S@desc
523    \let\EmFi@type\EmFi@temp
524  \else\ifx\EmFi@temp\EmFi@S@moddate
525    \let\EmFi@type\EmFi@temp
526  \else\ifx\EmFi@temp\EmFi@S@creationdate
527    \let\EmFi@type\EmFi@temp
528  \else\ifx\EmFi@temp\EmFi@S@size
529    \let\EmFi@type\EmFi@temp
530  \else
531    \EmFi@Error{%
532      Unknown type '\EmFi@temp'.\MessageBreak
533      Supported types: 'text', 'date', 'number', 'file',\MessageBreak
534      'desc', 'moddate', 'creationdate', 'size'%
535    }%
536  \fi\fi\fi\fi\fi\fi\fi\fi
537 }

538 \kv@define@key{EmFiFi}{title}{%
539  \def\EmFi@title{#1}%
540 }
```

`\EmFi@setboolean`

```
541 \def\EmFi@setboolean#1#2{%
542   \EdefSanitize\EmFi@temp{#2}%
543   \ifx\EmFi@temp\EmFi@S@true
544     \csname EmFi@#1true\endcsname
545   \else
546     \ifx\EmFi@temp\EmFi@S@false
547       \csname EmFi@#1false\endcsname
548     \else
549       \EmFi@Error{%
550         Unknown value '\EmFi@temp' for key '#1'.\MessageBreak
551         Supported values: 'true', 'false'%
552       }\@ehc
553     \fi
554   \fi
555 }
```

```
556 \kv@define@key{EmFiFi}{visible}[true]{%
557   \EmFi@setboolean{visible}{#1}%
558 }
```

```
559 \kv@define@key{EmFiFi}{edit}[true]{%
560   \EmFi@setboolean{edit}{#1}%
561 }
```

`\EmFi@sortkeys`

```
562 \def\EmFi@sortkeys{}
```

`\EmFi@sortorders`

```
563 \def\EmFi@sortorders{}
```

`\embedfilesort`

```
564 \def\embedfilesort{%
565   \kvsetkeys{EmFiSo}%
566 }
```

2.8   Embed the file

`\embedfile`

```
567 \def\embedfile{%
568   \ltx@ifnextchar[\EmFi@embedfile{\EmFi@embedfile[]}%
569 }
```

`\EmFi@embedfile`

```
570 \def\EmFi@embedfile[#1]#2{%
571   \ifEmFi@finished
572     \EmFi@Error{%
573       \string\embedfile\ltx@space after \string\embedfilefinish
574     }{%
575       The list of embedded files is already written.%
576     }%
577   \else
578     \begingroup
579       \def\EmFi@file{#2}%
580       \kvsetkeys{EmFi}{#1}%
581       \expandafter\expandafter\expandafter
582       \ifx\expandafter\expandafter\expandafter
583         \\\pdf@filesize{\EmFi@file}\\%
584       \EmFi@Error{%
585         File '\EmFi@file' not found%
586       }{%
```

```
587            The unknown file is not embedded.%
588          }%
589        \else
590          \edef\EmFi@@filespec{%
591            \pdf@escapestring{\EmFi@filespec}%
592          }%
593          \ifx\EmFi@ucfilespec\ltx@empty
594            \let\EmFi@@ucfilespec\ltx@empty
595          \else
596            \EmFi@convert\EmFi@ucfilespec\EmFi@@ucfilespec
597          \fi
598          \ifx\EmFi@desc\ltx@empty
599            \let\EmFi@@desc\ltx@empty
600          \else
601            \EmFi@convert\EmFi@desc\EmFi@@desc
602          \fi
603          \ifEmFi@item
604            \let\do\EmFi@do
605            \immediate\pdfobj{%
606              <<%
607                \EmFi@fieldlist
608              >>%
609            }%
610            \edef\EmFi@ci{\the\pdflastobj}%
611          \fi
612          \immediate\pdfobj stream attr{%
613            /Type/EmbeddedFile%
614            \ifx\EmFi@mimetype\ltx@empty
615            \else
616              /Subtype/\pdf@escapename{\EmFi@mimetype}%
617            \fi
618            /Params<<%
619              /ModDate(\pdf@filemoddate{\EmFi@file})%
620              /Size \pdf@filesize{\EmFi@file}%
621              /CheckSum<\pdf@filemdfivesum{\EmFi@file}>%
622            >>%
623          }file{\EmFi@file}\relax
624          \EmFi@defobj{EmbeddedFile}%
625          \immediate\pdfobj{%
626            <<%
627              /Type/Filespec%
628              \ifx\EmFi@filesystem\ltx@empty
629              \else
630              /FS/\pdf@escapename{\EmFi@filesystem}%
631              \fi
632              /F(\EmFi@@filespec)%
633              \ifx\EmFi@@ucfilespec\ltx@empty
634              \else
635                /UF(\EmFi@@ucfilespec)%
636              \fi
637              \ifx\EmFi@@desc\ltx@empty
638              \else
639                /Desc(\EmFi@@desc)%
640              \fi
641              /EF<<%
642                /F \the\pdflastobj\ltx@space 0 R%
643              >>%
644              \ifEmFi@item
645                /CI \EmFi@ci\ltx@space 0 R%
646              \fi
647            >>%
648          }%
```

```
649      \EmFi@defobj{Filespec}%
650      \EmFi@add{%
651        \EmFi@@filespec
652      }{\the\pdflastobj\ltx@space 0 R}%
653    \fi
654  \endgroup
655 \fi
656 }
```

\EmFi@do

```
657 \def\EmFi@do#1{%
658  \expandafter\ifx\csname EmFi@P@#1\endcsname\relax
659    \expandafter\ifx\csname EmFi@V@#1\endcsname\relax
660    \else
661      /\pdf@escapename{#1}\csname EmFi@V@#1\endcsname
662    \fi
663  \else
664    /\pdf@escapename{#1}<<%
665      \expandafter\ifx\csname EmFi@V@#1\endcsname\relax
666      \else
667        /D\csname EmFi@V@#1\endcsname
668      \fi
669      /P(\csname EmFi@P@#1\endcsname)%
670    >>%
671  \fi
672 }
```

\EmFi@convert

```
673 \def\EmFi@convert#1#2{%
674  \ifnum\pdf@strcmp{\EmFi@stringmethod}{psd}=0 %
675    \pdfstringdef\EmFi@temp{#1}%
676    \let#2\EmFi@temp
677  \else
678    \edef#2{\pdf@escapestring{#1}}%
679  \fi
680 }
```

```
681 \global\let\EmFi@list\ltx@empty
```

\EmFi@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,
            . . .).

```
682 \def\EmFi@add#1#2{%
683  \begingroup
684    \ifx\EmFi@list\ltx@empty
685      \xdef\EmFi@list{\noexpand\do{#1}{#2}}%
686    \else
687      \def\do##1##2{%
688        \ifnum\pdf@strcmp{##1}{#1}>0 %
689          \edef\x{%
690            \toks@{%
691              \the\toks@%
692              \noexpand\do{#1}{#2}%
693              \noexpand\do{##1}{##2}%
694            }%
695          }%
696          \x
697          \def\do####1####2{%
698            \toks@\expandafter{\the\toks@\do{####1}{####2}}%
699          }%
700          \def\stop{%
```

19

```
701        \xdef\EmFi@list{\the\toks@}%
702      }%
703    \else
704      \toks@\expandafter{\the\toks@\do{##1}{##2}}%
705    \fi
706  }%
707  \def\stop{%
708    \xdef\EmFi@list{\the\toks@\noexpand\do{#1}{#2}}%
709  }%
710  \toks@{}%
711  \EmFi@list\stop
712  \fi
713  \endgroup
714 }
```

\embedfilefinish

```
715 \def\embedfilefinish{%
716  \ifEmFi@finished
717    \EmFi@Error{%
718      Too many invocations of \string\embedfilefinish
719    }{%
720      The list of embedded files is already written.%
721    }%
722  \else
723    \ifx\EmFi@list\ltx@empty
724    \else
```

Write /EmbeddedFiles entry.

```
725      \global\EmFi@finishedtrue
726      \begingroup
727        \def\do##1##2{%
728          (##1)##2%
729        }%
730        \immediate\pdfobj{%
731          <<%
732            /Names[\EmFi@list]%
733          >>%
734        }%
735        \pdfnames{%
736          /EmbeddedFiles \the\pdflastobj\ltx@space 0 R%
737        }%
738      \endgroup
```

Write collection objects.

```
739      \ifx\EmFi@initialfile\ltx@empty
740      \else
741        \EmFi@collectiontrue
742      \fi
743      \ifEmFi@collection
744        \ifx\EmFi@initialfile\ltx@empty
745          \let\EmFi@@initialfile\ltx@empty
746        \else
747          \edef\EmFi@@initialfile{%
748            \pdf@escapestring{\EmFi@initialfile}%
749          }%
750        \fi
```

Look for initial file among the embedded files.

```
751        \begingroup
752          \let\f=N%
753          \def\do##1##2{%
754            \def\x{##1}%
755            \ifx\x\EmFi@@initialfile
756              \let\f=Y%
```

```
757              \let\do\ltx@gobbletwo
758            \fi
759          }%
760          \EmFi@list
761        \expandafter\endgroup
762        \ifx\f Y%
763        \else
764          \@PackageWarningNoLine{embedfile}{%
765            Missing initial file '\EmFi@initialfile'\MessageBreak
766            among the embedded files%
767          }%
768          \let\EmFi@initialfile\ltx@empty
769          \let\EmFi@@initialfile\ltx@empty
770        \fi
771        \ifcase\EmFi@sortcase
772          \def\EmFi@temp{}%
773        \or
774          \def\EmFi@temp{%
775            /S\EmFi@sortkeys
776            /A \EmFi@sortorders
777          }%
778        \else
779          \def\EmFi@temp{%
780            /S[\EmFi@sortkeys]%
781            /A[\EmFi@sortorders]%
782          }%
783        \fi
784        \def\EmFi@@order##1{%
785          \ifnum\EmFi@order>1 %
786            /O ##1%
787          \fi
788        }%
789        \immediate\pdfobj{%
790          <<%
791            \ifx\EmFi@schema\ltx@empty
792            \else
793              /Schema<<\EmFi@schema>>%
794            \fi
795            \ifx\EmFi@@initialfile\ltx@empty
796            \else
797              /D(\EmFi@@initialfile)%
798            \fi
799            \ifx\EmFi@view\EmFi@S@tile
800              /View/T%
801            \else\ifx\EmFi@view\EmFi@S@hidden
802              /View/H%
803            \fi\fi
804            \ifx\EmFi@temp\ltx@empty
805              \EmFi@temp
806            \else
807              /Sort<<\EmFi@temp>>%
808            \fi
809          >>%
810        }%
811        \pdfcatalog{%
812          /Collection \the\pdflastobj\ltx@space0 R%
813        }%
814      \fi
815    \fi
816  \fi
817 }

818 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```
819 \expandafter\ifx\csname AtEndDocument\endcsname\relax
820 \else
821   \AtEndDocument{\embedfilefinish}%
822 \fi

823 \EmFi@AtEnd%
824 ⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

```
825 ⟨*test1⟩

826 \catcode'\{=1 %
827 \catcode'\}=2 %
828 \catcode'\#=6 %
829 \catcode'\@=11 %
830 \expandafter\ifx\csname count@\endcsname\relax
831   \countdef\count@=255 %
832 \fi
833 \expandafter\ifx\csname @gobble\endcsname\relax
834   \long\def\@gobble#1{}%
835 \fi
836 \expandafter\ifx\csname @firstofone\endcsname\relax
837   \long\def\@firstofone#1{#1}%
838 \fi
839 \expandafter\ifx\csname loop\endcsname\relax
840   \expandafter\@firstofone
841 \else
842   \expandafter\@gobble
843 \fi
844 {%
845   \def\loop#1\repeat{%
846     \def\body{#1}%
847     \iterate
848   }%
849   \def\iterate{%
850     \body
851       \let\next\iterate
852     \else
853       \let\next\relax
854     \fi
855     \next
856   }%
857   \let\repeat=\fi
858 }%
859 \def\RestoreCatcodes{}
860 \count@=0 %
861 \loop
862   \edef\RestoreCatcodes{%
863     \RestoreCatcodes
864     \catcode\the\count@=\the\catcode\count@\relax
865   }%
866 \ifnum\count@<255 %
867   \advance\count@ 1 %
868 \repeat
869
870 \def\RangeCatcodeInvalid#1#2{%
871   \count@=#1\relax
872   \loop
873     \catcode\count@=15 %
874   \ifnum\count@<#2\relax
```

```
875     \advance\count@ 1 %
876   \repeat
877 }
878 \def\RangeCatcodeCheck#1#2#3{%
879   \count@=#1\relax
880   \loop
881     \ifnum#3=\catcode\count@
882     \else
883       \errmessage{%
884         Character \the\count@\space
885         with wrong catcode \the\catcode\count@\space
886         instead of \number#3%
887       }%
888     \fi
889   \ifnum\count@<#2\relax
890     \advance\count@ 1 %
891   \repeat
892 }
893 \def\space{ }
894 \expandafter\ifx\csname LoadCommand\endcsname\relax
895   \def\LoadCommand{\input embedfile.sty\relax}%
896 \fi
897 \def\Test{%
898   \RangeCatcodeInvalid{0}{47}%
899   \RangeCatcodeInvalid{58}{64}%
900   \RangeCatcodeInvalid{91}{96}%
901   \RangeCatcodeInvalid{123}{255}%
902   \catcode`\@=12 %
903   \catcode`\\=0 %
904   \catcode`\%=14 %
905   \LoadCommand
906   \RangeCatcodeCheck{0}{36}{15}%
907   \RangeCatcodeCheck{37}{37}{14}%
908   \RangeCatcodeCheck{38}{47}{15}%
909   \RangeCatcodeCheck{48}{57}{12}%
910   \RangeCatcodeCheck{58}{63}{15}%
911   \RangeCatcodeCheck{64}{64}{12}%
912   \RangeCatcodeCheck{65}{90}{11}%
913   \RangeCatcodeCheck{91}{91}{15}%
914   \RangeCatcodeCheck{92}{92}{0}%
915   \RangeCatcodeCheck{93}{96}{15}%
916   \RangeCatcodeCheck{97}{122}{11}%
917   \RangeCatcodeCheck{123}{255}{15}%
918   \RestoreCatcodes
919 }
920 \Test
921 \csname @@end\endcsname
922 \end
923 ⟨/test1⟩
```

3.2   Simple test

```
924 ⟨*test2⟩
925 \input embedfile.sty\relax
926 \embedfile[%
927   stringmethod=escape,%
928   mimetype=plain/text,%
929   desc={LaTeX docstrip source archive for package 'embedfile'},%
930   id={embedfile.dtx}%
931 ]{embedfile.dtx}
932 \nopagenumbers
933 Test (plain-\TeX): {\tt embedfile.dtx} should be embedded.%
934
```

```
935 \def\Test#1{%
936   \par
937   \embedfileifobjectexists{embedfile.dtx}{#1}{%
938     Object #1 (embedfile.dtx): %
939     \embedfilegetobject{embedfile.dtx}{#1}%
940   }{%
941     \errmessage{Missing object #1 (embedfile.dtx)}%
942   }%
943 }
944 \Test{EmbeddedFile}
945 \Test{Filespec}
946 \embedfilefinish
947 \bye
948 ⟨/test2⟩
949 ⟨*test3⟩
950 \NeedsTeXFormat{LaTeX2e}
951 \let\SavedJobname\jobname
952 \def\jobname{embedfile}
953 \RequirePackage{dtx-attach}[2011/04/13]
954 \let\jobname\SavedJobname
955 \documentclass{minimal}
956 \begin{document}
957   Test (\LaTeX): \texttt{embedfile.dtx} should be embedded.%
958 \end{document}
959 ⟨/test3⟩
```

3.3   Test for ini-TeX

```
960 ⟨*test4⟩
961 \catcode`\{=1 %
962 \catcode`\}=2 %
963 \input ifluatex.sty %
964 \ifluatex
965   \directlua{%
966     tex.enableprimitives('', {%
967       'pdflastobj',%
968       'pdfnames',%
969       'pdfobj',%
970       'pdfoutput'%
971     })%
972   }%
973 \fi
974 \pdfoutput=1 %
975 \input embedfile.sty %
976 \shipout\hbox{}
977 \embedfile[%
978   stringmethod=escape,%
979   mimetype=plain/text,%
980   desc={iniTeX source},%
981 ]{\jobname.tex}
982 \embedfilefinish
983 \end
984 ⟨/test4⟩
```

4   Installation

4.1   Download

Package.   This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/embedfile.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/embedfile.pdf Documentation.

---

[1] ftp://ftp.ctan.org/tex-archive/

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:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TeX 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 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):

```
embedfile.sty                      → tex/generic/oberdiek/embedfile.sty
dtx-attach.sty                     → tex/generic/oberdiek/dtx-attach.sty
embedfile.pdf                      → doc/latex/oberdiek/embedfile.pdf
embedfile-example-plain.tex        → doc/latex/oberdiek/embedfile-example-plain.tex
embedfile-example-collection.tex   → doc/latex/oberdiek/embedfile-example-collection.tex
test/embedfile-test1.tex           → doc/latex/oberdiek/test/embedfile-test1.tex
test/embedfile-test2.tex           → doc/latex/oberdiek/test/embedfile-test2.tex
test/embedfile-test3.tex           → doc/latex/oberdiek/test/embedfile-test3.tex
test/embedfile-test4.tex           → doc/latex/oberdiek/test/embedfile-test4.tex
embedfile.dtx                      → source/latex/oberdiek/embedfile.dtx
```

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 TeX distribution (teTeX, mikTeX, ...) relies on file name databases, you must refresh these. For example, teTeX users run `texhash` or `mktexlsr`.

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 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 pdfLATEX:

```
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, Sixth Edition, Version 1.7*, Oktober 2006; http://www.adobe.com/devnet/pdf/pdf_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.

[2007/10/28 v2.0]

- Collection support added (PDF 1.7).

[2007/10/29 v2.1]

- Export of object references by adding new option id and new macros \embedfileifobjectexists and \embedfilegetobject.

[2007/11/11 v2.2]

- Use of package pdftexcmds for LuaTeX support.

[2007/11/25 v2.3]

- Fix in use of \pdf@filesize, bug introduced in previous version.

[2009/09/25 v2.4]

- Bug fix: If hyperref is used with option unicode, the Unicode encoded file name causes trouble. Therefore \pdfstringdef is now never used for option filespec, always method escape is applied (Peter Cibulka).

- Bug fix for initialfile.

- Bug fix for file names in /EmbeddedFiles.

- New option ucfilespec for file name support in Unicode (since PDF 1.7).

[2010/03/01 v2.5]

- Compatibility for ini-TeX.

- Package keyval replaced by packages kvsetkeys and kvdefinekeys because of compatibility for ini-TeX.

- TDS location moved from TDS:tex/latex/oberdiek/embedfile.sty to TDS:tex/generic/oberdiek/embedfile.sty.

[2011/04/13 v2.6]

- Docu fixes (thanks Hans-Martin Münch).

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; plain numbers refer to the code lines where the entry is used.

27

28

29