# The **accsupp** package

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

2018/03/28 v0.5

**Abstract**

Since PDF 1.5 portions of a page can be marked for better accessibility support. For example, replacement texts or expansions of abbreviations can be provided. Package accsupp starts with providing a minimal low-level interface for programmers. Status is experimental.

# Contents

---

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

# 1 Documentation

## 1.1 Macros

Section "10.8 Accessibility Support" of the PDF reference [1] lists some features that can be added by operators for marked content.

---

$\boxed{\texttt{\textbackslash BeginAccSupp} \; \{\langle options \rangle\}}$

It puts the operator BDC in the page stream:

```
/Span
<<…>>       % property dictionary
BDC
```

The contents of the dictionary is controlled by ⟨*options*⟩. See sections 1.1.1 and 1.1.2.

---

$\boxed{\texttt{\textbackslash EndAccSupp} \; \{\langle options \rangle\}}$

It puts the operator EMC in the page stream. The only option is pdfliteral, see section 1.3.1.

**Note:** The caller is responsible for the placement of \BeginAccSupp and \EndAccSupp pairs. Especially page breaks are not allowed in between.

### 1.1.1 Feature options

The PDF reference [1] describes and explains the different features. The names of the feature options are the same as the key names for the property dictionary for operator BDC, see \BeginAccSupp.

**ActualText:** Provides a replacement text, see examples in section 2.

**Alt:** Provides an alternate description.

**E:** Provides the expansion of an abbreviation or an acronym.

**Lang:** Specifies the language.

### 1.1.2 Input methods

Except for Lang option method controls how the argument for ActualText, Alt, and E are interpreted.

**method=plain:** The string is only expanded and written without further treatment. Special characters are not protected, thus this method may result in an invalid PDF file.

**method=escape:** The string is expanded and special characters are escaped. The result is a valid PDF string.

**method=hex:** The string is given in hexadecimal notation. Section 2.1 shows an example.

**method=pdfstringdef:** If package hyperref is loaded, then its `\pdfstringdef` is used. This method is slow, but useful if the string contains arbitrary TeX code.

**unicode:** This option is needed, if the string is given as Unicode string (16 bit). Internally it adds the string prefix for Unicode. In case of method=pdfstringdef it passes the option to `\hypersetup`.

## 1.2 Workaround, option **space**

PDF specification says in "10.8.3 Replacement Text"

> Just as alternative descriptions can be provided for images and other items that do not translate naturally into text (…), replacement text can be specified for content that does translate into text but that is represented in a nonstandard way. These nonstandard representations might include, for example, glyphs for ligatures or custom characters, or inline graphics corresponding to letters in an illuminated manustript or to dropped capitals.

However, the ActualText is ignored in Acrobat Reader (until version 9 at least), if the content does not contain glyphs. Option space adds such an invisible glyph, a space character. The font name can be configured by option spacefont, the default font is `phvr8r`. The character slot is given by option spacechar, default is 32, the usual position of the space character.

These options can also be given as package options or in macro `\AccSuppSetup` that takes a key value list as argument. Usually only option space is necessary, if the contents does not contain glyphs otherwise. Option space is enabled by space or space=true and disabled by space=false. It is disabled as default. The option is evaluated by `\BeginAccSupp` and ignored by `\EndAccSupp`.

Note: Even with option space I get sometimes wrong texts when cut & paste from AR7/Linux or AR8/Linux, e.g.

```
Hello → Helo, Helol, Hell, …
Hello World → Helo WorldW, Helo World, …
```

I do not know what Acrobat Reader is doing here, thus feedback and insights are welcome.

## 1.3 Driver options

Driver options are package options only. The special TeX compilers pdfTeX, LuaTeX and XeTeX are detected automatically. The default for unrecognized drivers can be set by defining `\ActualTextDriverDefault`. This can be done in the configuration file `accsupp.cfg`.

### 1.3.1 Option **pdftex**

Package option pdftex is used for pdfTeX in PDF mode. Additionally \BeginAcc-Supp and \EndAccSupp understand option pdfliteral. It controls the modifier keyword for \pdfliteral:

pdfliteral=direct ⇒ \pdfliteral direct{...}

### 1.3.2 Option **luatex**

Package option luatex is used for LuaTeX in PDF mode. Additionally \BeginAcc-Supp and \EndAccSupp understand option pdfliteral. It controls the modifier keyword for \pdfextension literal:

pdfliteral=direct ⇒ \pdfextension literal direct{...}

### 1.3.3 Option **dvipdfm**

Package option dvipdfm and its aliases dvipdfmx xetex are used for drivers that support dvipdfm specials.

### 1.3.4 Option **dvips**

Package option dvips and its alias dvipsone write pdfmark specials in the output. Unhappily these pdfmark operators are ignored by ghostscript (latest tested version is 8.54). Perhaps they are recognized by commercial distiller applications.

### 1.3.5 Turning off page stream compression

For debugging it is useful to have uncompressed page stream objects. This can be done afterwards via pdftk:

pdftk file.pdf output file-uncompressed.pdf uncompress

Or the PDF file is generated uncompressed:

**pdfTeX:** \pdfcompresslevel=0

**dvipdfm:** dvipdfm -z0 ...

**dvipdfmx:** dvipdfmx -z0 ...

**ghostscript/ps2pdf:** ps2pdf -dCompressPages=false input.ps output.pdf

## 2 Example

## 2.1 Example \notparallel

```
 1 ⟨*example1⟩

 2 %<<END
 3 \documentclass{article}
 4 \usepackage{accsupp}[2007/11/14]
 5 \usepackage{centernot}
 6 % U+2226 NOT PARALLEL
 7 % \mathrel{...} prevents page break in between
 8 \newcommand*\notparallel}{%
 9  \ensuremath{%
10   \mathrel{%
11    \BeginAccSupp{method=hex,unicode,ActualText=2226}%
12    \centernot{\parallel}%
13    \EndAccSupp{}%
14   }%
15  }%
```

```
16 }
17 \begin{document}
18 \begin{equation}
19 A\notparallel B
20 \end{equation}
21 \end{document}
22 %END
```

23 ⟨/example1⟩

## 2.2 Example with **pdfstringdef**

24 ⟨*example2⟩

```
25 %<<END
26 \documentclass{article}
27 \usepackage[unicode]{hyperref}
28 \usepackage{accsupp}[2007/11/14]
29 \begin{document}
30  \begin{equation}
31   \BeginAccSupp{
32    method=pdfstringdef,
33    unicode,
34    ActualText={%
35     a\texttwosuperior +b\texttwosuperior
36     =c\texttwosuperior
37    }
38   }
39   a^2 + b^2 = c^2
40   \EndAccSupp{}
41  \end{equation}
42 \end{document}
43 %END
```

44 ⟨/example2⟩

# 3 Implementation

## 3.1 Package

45 ⟨*package⟩

```
46 \begingroup\catcode61\catcode48\catcode32=10\relax%
47  \catcode13=5 % ^^M
48  \endlinechar=13 %
49  \catcode123=1 % {
50  \catcode125=2 % }
51  \catcode64=11 % @
52  \def\x{\endgroup
53   \expandafter\edef\csname ACCSUPP@AtEnd\endcsname{%
54    \endlinechar=\the\endlinechar\relax
55    \catcode13=\the\catcode13\relax
56    \catcode32=\the\catcode32\relax
57    \catcode35=\the\catcode35\relax
58    \catcode61=\the\catcode61\relax
59    \catcode64=\the\catcode64\relax
60    \catcode123=\the\catcode123\relax
61    \catcode125=\the\catcode125\relax
62   }%
63  }%
64 \x\catcode61\catcode48\catcode32=10\relax%
65 \catcode13=5 % ^^M
66 \endlinechar=13 %
67 \catcode35=6 % #
68 \catcode64=11 % @
69 \catcode123=1 % {
```

```
70 \catcode125=2 % }
71 \def\TMP@EnsureCode#1#2{%
72   \edef\ACCSUPP@AtEnd{%
73     \ACCSUPP@AtEnd
74     \catcode#1=\the\catcode#1\relax
75   }%
76   \catcode#1=#2\relax
77 }
78 \TMP@EnsureCode{10}{12}% ^^J
79 \TMP@EnsureCode{33}{12}% !
80 \TMP@EnsureCode{39}{12}% '
81 \TMP@EnsureCode{40}{12}% (
82 \TMP@EnsureCode{41}{12}% )
83 \TMP@EnsureCode{42}{12}% *
84 \TMP@EnsureCode{44}{12}% ,
85 \TMP@EnsureCode{45}{12}% -
86 \TMP@EnsureCode{46}{12}% .
87 \TMP@EnsureCode{47}{12}% /
88 \TMP@EnsureCode{58}{12}% :
89 \TMP@EnsureCode{60}{12}% <
90 \TMP@EnsureCode{62}{12}% >
91 \TMP@EnsureCode{94}{7}% ^ (superscript)
92 \TMP@EnsureCode{96}{12}% `
93 \TMP@EnsureCode{254}{12}% ^^fe
94 \TMP@EnsureCode{255}{12}% ^^ff
95 \edef\ACCSUPP@AtEnd{\ACCSUPP@AtEnd\noexpand\endinput}
```

Package identification.

```
96 \NeedsTeXFormat{LaTeX2e}
97 \ProvidesPackage{accsupp}%
98   [2018/03/28 v0.5 Accessibility support by marked content (HO)]
99 \RequirePackage{pdfescape}[2007/02/25]
100 \RequirePackage{ifpdf}
101 \RequirePackage{ifxetex}
102 \RequirePackage{kvoptions}

103 \SetupKeyvalOptions{%
104   family=ACCSUPP,%
105   prefix=ACCSUPP@%
106 }
```

## 3.2  Space setup

```
107 \DeclareBoolOption{space}
108 \DeclareStringOption[phvr8r]{spacefont}
109 \DeclareStringOption[32]{spacechar}
```

## 3.3  Driver detection and setup

**Driver declarations.**

```
110 \def\ACCSUPP@DefineDriverKey{%
111   \@dblarg\ACCSUPP@@DefineDriverKey
112 }
113 \def\ACCSUPP@@DefineDriverKey[#1]#2{%
114   \define@key{ACCSUPP}{#2}[]{%
115     \def\ACCSUPP@driver{#1}%
116   }%
117   \g@addto@macro\ACCSUPP@DisableOptions{%
118     \DisableKeyvalOption{ACCSUPP}{#2}%
119   }%
120 }
121 \let\ACCSUPP@DisableOptions\@empty
122 \ACCSUPP@DefineDriverKey{pdftex}
123 \ACCSUPP@DefineDriverKey{luatex}
124 \ACCSUPP@DefineDriverKey{dvips}
```

```
125 \ACCSUPP@DefineDriverKey[dvips]{dvipsone}
126 \ACCSUPP@DefineDriverKey{dvipdfm}
127 \ACCSUPP@DefineDriverKey[dvipdfm]{dvipdfmx}
128 \ACCSUPP@DefineDriverKey[dvipdfm]{xetex}
129 \let\ACCSUPP@driver\relax
130 \InputIfFileExists{accsupp.cfg}{}{}
131 \providecommand*{\ActualTextDriverDefault}{dvips}
132 \ifpdf
133   \ifx\pdfextension\@undefined
134     \def\ACCSUPP@driver{pdftex}%
135   \else
136     \def\ACCSUPP@driver{luatex}%
137   \fi
138 \else
139   \ifxetex
140     \def\ACCSUPP@driver{dvipdfm}%
141   \else
142     \ifx\ACCSUPP@driver\relax
143       \let\ACCSUPP@driver\ActualTextDriverDefault
144     \fi
145   \fi
146 \fi
```

**Process options.**

```
147 \ProcessKeyvalOptions*
148 \ACCSUPP@DisableOptions
```

**Driver validation and loading.**

```
149 \def\ACCSUPP@temp{pdftex}%
150 \ifpdf
151   \ifx\pdfextension\@undefined\else
152     \def\ACCSUPP@temp{luatex}%
153   \fi
154   \ifx\ACCSUPP@temp\ACCSUPP@driver
155   \else
156     \PackageWarningNoLine{accsupp}{%
157       Wrong driver `\ACCSUPP@driver', using `pdftex' instead%
158     }%
159 ed as
160     \let\ACCSUPP@driver\ACCSUPP@temp
161   \fi
162 \else
163   \ifx\ACCSUPP@temp\ACCSUPP@driver
164     \PackageError{accsupp}{%
165       Wrong driver, pdfTeX is not running in PDF mode.\MessageBreak
166       Package loading is aborted%
167     }\@ehc
168     \expandafter\expandafter\expandafter\ACCSUPP@AtEnd
169   \fi
170   \def\ACCSUPP@temp{dvipdfm}%
171   \ifxetex
172     \ifx\ACCSUPP@temp\ACCSUPP@driver
173     \else
174       \PackageWarningNoLine{accsupp}{%
175         Wrong driver `\ACCSUPP@driver',\MessageBreak
176         using `dvipdfm' for XeTeX instead%
177       }%
178       \let\ACCSUPP@driver\ACCSUPP@temp
179     \fi
180   \fi
181 \fi%
```

```
182 \ifx\ACCSUPP@driver\relax
183  \PackageError{accsupp}{%
184    Missing driver option.\MessageBreak
185    Package loading is aborted%
186  }\@ehc
187  \expandafter\ACCSUPP@AtEnd
188 \fi%
189 \InputIfFileExists{accsupp-\ACCSUPP@driver.def}{}{%
190  \PackageError{accsupp}{%
191    Unsupported driver `\ACCSUPP@driver'.\MessageBreak
192    Package loading is aborted%
193  }\@ehc
194  \ACCSUPP@AtEnd
195 }%
```

## 3.4   Main macro

```
196 \DeclareBoolOption{unicode}
197 \DeclareStringOption[page]{pdfliteral}
198 \DeclareStringOption{Lang}
199 \def\ACCSUPP@method{escape}
200 \define@key{ACCSUPP}{method}{%
201  \@ifundefined{ACCSUPP@method@#1}{%
202    \PackageError{accsupp}{%
203      Ignoring unknown method `#1'%
204    }\@ehc
205  }{%
206    \edef\ACCSUPP@method{#1}%
207  }%
208 }
209 \let\ACCSUPP@Lang\relax
210 \def\ACCSUPP@temp#1{%
211  \expandafter\ACCSUPP@@temp\csname ACCSUPP@#1\endcsname{#1}%
212 }
213 \def\ACCSUPP@@temp#1#2{%
214  \let#1\relax
215  \define@key{ACCSUPP}{#2}{%
216    \def#1{##1}%
217    \ifx#1\@empty
218      \def#1{()}%
219    \else
220      \csname ACCSUPP@method@\ACCSUPP@method\endcsname#1%
221    \fi
222  }%
223 }
224 \ACCSUPP@temp{Alt}
225 \ACCSUPP@temp{ActualText}
226 \ACCSUPP@temp{E}

227 \newcommand*{\BeginAccSupp}[1]{%
228  \begingroup
229    \setkeys{ACCSUPP}{#1}%
230    \edef\ACCSUPP@span{%
231      /Span<<%
232        \ifx\ACCSUPP@Lang\relax
233        \else
234          /Lang\ACCSUPP@Lang
235        \fi
236        \ifx\ACCSUPP@Alt\relax
237        \else
238          /Alt\ACCSUPP@Alt
239        \fi
240        \ifx\ACCSUPP@ActualText\relax
```

```
241    \else
242      /ActualText\ACCSUPP@ActualText
243    \fi
244    \ifx\ACCSUPP@E\relax
245    \else
246      /E\ACCSUPP@E
247    \fi
248    >>%
249  }%
250  \ACCSUPP@bdc
251  \ACCSUPP@space
252  \endgroup
253 }
254 \newcommand*{\EndAccSupp}[1]{%
255  \begingroup
256    \setkeys{ACCSUPP}{#1}%
257    \ACCSUPP@emc
258  \endgroup
259 }
260 \def\ACCSUPP@space{%
261  \ifACCSUPP@space
262    \begingroup
263      \@ifundefined{ACCSUPP@Font}{%
264        \global\font\ACCSUPP@Font=\ACCSUPP@spacefont\relax
265      }{}%
266      \leavevmode
267      \setbox\z@\hbox{\ACCSUPP@Font\char\ACCSUPP@spacechar}%
268      \wd\z@\z@
269      \ht\z@\z@
270      \dp\z@\z@
271      \copy\z@
272    \endgroup
273  \fi
274 }
275 \newcommand*{\AccSuppSetup}{%
276  \setkeys{ACCSUPP}%
277 }
```

### 3.4.1 Input methods

```
278 \def\ACCSUPP@method@plain#1{%
279  \csname @safe@activestrue\endcsname
280  \edef#1{%
281    (%
282    \ifACCSUPP@unicode
283      \string\376\string\377%
284    \fi
285    #1%
286    )%
287  }%
288  \@onelevel@sanitize#1%
289 }

290 \def\ACCSUPP@method@escape#1{%
291  \EdefEscapeString#1{%
292    \ifACCSUPP@unicode
293      ^^fe^^ff%
294    \fi
295    #1%
296  }%
297  \edef#1{(#1)}%
298 }%

299 \def\ACCSUPP@method@hex#1{%
300  \edef#1{%
```

```
301    <%
302    \ifACCSUPP@unicode
303      FEFF%
304    \fi
305    #1%
306    >%
307  }%
308 }
309 \def\ACCSUPP@method@pdfstringdef#1{%
310   \ifACCSUPP@unicode
311     \@ifundefined{hypersetup}{}{%
312       \hypersetup{unicode}%
313     }%
314   \fi
315   \@ifundefined{pdfstringdef}{%
316     \PackageError{accsupp}{%
317       Method `pdfstringdef' requires package `hyperref'%
318     }\@ehc
319     \let\ACCSUPP@temp\@empty
320   }{%
321     \begingroup
322       \setbox0=\hbox{%
323         \pdfstringdef\ACCSUPP@temp#1%
324         \global\let\ACCSUPP@temp\ACCSUPP@temp
325       }%
326     \endgroup
327   }%
328   \edef#1{(\ACCSUPP@temp)}%
329 }
330 \ACCSUPP@AtEnd%
331 ⟨/package⟩
```

## 3.5  Drivers

### 3.5.1  Driver **pdftex**

```
332 ⟨*pdftex⟩
333 \NeedsTeXFormat{LaTeX2e}
334 \ProvidesFile{accsupp-pdftex.def}%
335   [2018/03/28 v0.5 accsupp driver for pdfTeX (HO)]%
336 \def\ACCSUPP@bdc{%
337   \pdfliteral\ACCSUPP@pdfliteral{\ACCSUPP@span BDC}%
338 }
339 \def\ACCSUPP@emc{%
340   \pdfliteral\ACCSUPP@pdfliteral{EMC}%
341 }
342 ⟨/pdftex⟩
```

### 3.5.2  Driver **luatex**

```
343 ⟨*luatex⟩
344 \NeedsTeXFormat{LaTeX2e}
345 \ProvidesFile{accsupp-luatex.def}%
346   [2018/03/28 v0.5 accsupp driver for pdfTeX (HO)]%
347 \protected\def\ACCSUPP@pdf@literal{\pdfextension literal}
348 \def\ACCSUPP@bdc{%
349   \ACCSUPP@pdf@literal\ACCSUPP@pdfliteral{\ACCSUPP@span BDC}%
350 }
351 \def\ACCSUPP@emc{%
352   \ACCSUPP@pdf@literal\ACCSUPP@pdfliteral{EMC}%
353 }
354 ⟨/luatex⟩
```

### 3.5.3 Driver **dvipdfm**

355 ⟨*dvipdfm⟩
356 \NeedsTeXFormat{LaTeX2e}
357 \ProvidesFile{accsupp-dvipdfm.def}%
358   [2018/03/28 v0.5 accsupp driver for dvipdfm (HO)]%

359 \def\ACCSUPP@bdc{%
360   \special{pdf:content \ACCSUPP@span BDC}%
361 }
362 \def\ACCSUPP@emc{%
363   \special{pdf:content EMC}%
364 }

365 ⟨/dvipdfm⟩

### 3.5.4 Driver **dvips**

366 ⟨*dvips⟩
367 \NeedsTeXFormat{LaTeX2e}
368 \ProvidesFile{accsupp-dvips.def}%
369   [2018/03/28 v0.5 accsupp driver for dvips (HO)]%

370 \def\ACCSUPP@bdc{%
371   \special{ps:[\ACCSUPP@span/BDC pdfmark}%
372 }
373 \def\ACCSUPP@emc{%
374   \special{ps:[/EMC pdfmark}%
375 }

376 ⟨/dvips⟩

## 4  Test

## 4.1  Catcode checks for loading

377 ⟨*test1⟩

378 \NeedsTeXFormat{LaTeX2e}
379 \documentclass{minimal}
380 \makeatletter
381 \def\RestoreCatcodes{}
382 \count@=0 %
383 \loop
384   \edef\RestoreCatcodes{%
385     \RestoreCatcodes
386     \catcode\the\count@=\the\catcode\count@\relax
387   }%
388 \ifnum\count@<255 %
389   \advance\count@\@ne
390 \repeat
391
392 \def\RangeCatcodeInvalid#1#2{%
393   \count@=#1\relax
394   \loop
395     \catcode\count@=15 %
396   \ifnum\count@<#2\relax
397     \advance\count@\@ne
398   \repeat
399 }
400 \def\Test{%
401   \RangeCatcodeInvalid{0}{47}%
402   \RangeCatcodeInvalid{58}{64}%
403   \RangeCatcodeInvalid{91}{96}%
404   \RangeCatcodeInvalid{123}{127}%
405   \catcode`\@=12 %
406   \catcode`\\=0 %

```
407  \catcode`\{=1 %
408  \catcode`\}=2 %
409  \catcode`\#=6 %
410  \catcode`\[=12 %
411  \catcode`\]=12 %
412  \catcode`\%=14 %
413  \catcode`\ =10 %
414  \catcode13=5 %
415  \RequirePackage{accsupp}[2018/03/28]\relax
416  \RestoreCatcodes
417 }
418 \Test
419 \csname @@end\endcsname
420 \end

421 ⟨/test1⟩
```

# 5  Installation

## 5.1  Download

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

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

**CTAN:macros/latex/contrib/oberdiek/accsupp.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: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.

## 5.2  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/

## 5.3  Package installation

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

    tex accsupp.dtx

---

[1] http://ctan.org/pkg/accsupp

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

```
accsupp.sty            → tex/latex/oberdiek/accsupp.sty
accsupp-pdftex.def     → tex/latex/oberdiek/accsupp-pdftex.def
accsupp-luatex.def     → tex/latex/oberdiek/accsupp-luatex.def
accsupp-dvipdfm.def    → tex/latex/oberdiek/accsupp-dvipdfm.def
accsupp-dvips.def      → tex/latex/oberdiek/accsupp-dvips.def
accsupp.pdf            → doc/latex/oberdiek/accsupp.pdf
accsupp-example1.tex   → doc/latex/oberdiek/accsupp-example1.tex
accsupp-example2.tex   → doc/latex/oberdiek/accsupp-example2.tex
test/accsupp-test1.tex → doc/latex/oberdiek/test/accsupp-test1.tex
accsupp.dtx            → source/latex/oberdiek/accsupp.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`.

## 5.4 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`.

## 5.5 Some details for the interested

**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{accsupp.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 accsupp.dtx
makeindex -s gind.ist accsupp.idx
pdflatex accsupp.dtx
makeindex -s gind.ist accsupp.idx
pdflatex accsupp.dtx
```

# 6 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `accsupp.xml`.

```
422 ⟨*catalogue⟩
423 <?xml version='1.0' encoding='us-ascii'?>
424 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
```

```
425 <entry datestamp='$Date$' modifier='$Author$' id='accsupp'>
426  <name>accsupp</name>
427  <caption>Better accessibility support for PDF files.</caption>
428  <authorref id='auth:oberdiek'/>
429  <copyright owner='Heiko Oberdiek' year='2007,2010'/>
430  <license type='lppl1.3'/>
431  <version number='0.4'/>
432  <description>
433    Since PDF 1.5 portions of a page can be marked for better accessibility
434    support. For example, replacement texts or expansions of abbreviations can
435    be provided. This package starts with providing a minimal low-level
436    interface for programmers; its status is experimental.
437    <p/>
438    The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
439  </description>
440  <documentation details='Package documentation'
441      href='ctan:/macros/latex/contrib/oberdiek/accsupp.pdf'/>
442  <ctan file='true' path='/macros/latex/contrib/oberdiek/accsupp.dtx'/>
443  <miktex location='oberdiek'/>
444  <texlive location='oberdiek'/>
445  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
446 </entry>
447 ⟨/catalogue⟩
```

# 7 References

[1] Adobe Systems Incorporated, *PDF Reference*, 6th edition, 2006. http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference.pdf

# 8 History

## [2007/03/21 v0.1]

- First version.

## [2007/11/14 v0.2]

- Various bug fixes.

- Catcode section rewritten, test added.

## [2010/01/16 v0.3]

- \AccSuppSetup added.

- Options space, spacefont, spacechar added.

## [2016/05/16 v0.4]

- Documentation updates.

## [2018/03/28 v0.5]

- LuaTeX support added.

# 9 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.