

The `accsupp` package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2010/01/16 v0.3

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

1	Documentation	2
1.1	Macros	2
1.1.1	Feature options	2
1.1.2	Input methods	2
1.2	Workaround, option <code>space</code>	3
1.3	Driver options	3
1.3.1	Option <code>pdftex</code>	3
1.3.2	Option <code>dvipdfm</code>	3
1.3.3	Option <code>dvips</code>	4
1.3.4	Turning off page stream compression	4
2	Example	4
2.1	Example <code>\notparallel</code>	4
2.2	Example with <code>pdfstringdef</code>	4
3	Implementation	5
3.1	Package	5
3.2	Space setup	6
3.3	Driver detection and setup	6
3.4	Main macro	7
3.4.1	Input methods	9
3.5	Drivers	9
3.5.1	Driver <code>pdftex</code>	9
3.5.2	Driver <code>dvipdfm</code>	10
3.5.3	Driver <code>dvips</code>	10
4	Test	10
4.1	Catcode checks for loading	10
5	Installation	11
5.1	Download	11
5.2	Bundle installation	11
5.3	Package installation	12
5.4	Refresh file name databases	12
5.5	Some details for the interested	12
6	References	13

7 History	13
[2007/03/21 v0.1]	13
[2007/11/14 v0.2]	13
[2010/01/16 v0.3]	13
8 Index	13

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.

`\BeginAccSupp {<options>}`

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.

`\EndAccSupp {<options>}`

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 manuscript 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 `pdf \TeX` and `Xe \TeX` 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 `pdf \TeX` in PDF mode. Additionally `\BeginAccSupp` and `\EndAccSupp` understand option `pdfliteral`. It controls the modifier keyword for `\pdfliteral`:

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

1.3.2 Option `dvipdfm`

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

1.3.3 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.

1.3.4 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\textttwosuperior +b\textttwosuperior
36     =c\textttwosuperior
37 }
38 }
39 a^2 + b^2 = c^2
40 \EndAccSupp{}
41 \end{equation}
42 \end{document}
43 %END
44 \end{example2}

```

3 Implementation

3.1 Package

```

45 (*package)
46 \begingroup
47 \catcode123 1 % {
48 \catcode125 2 % }
49 \def\x{\endgroup
50 \expandafter\edef\csname ACCSUPP@AtEnd\endcsname{%
51     \catcode35 \the\catcode35\relax
52     \catcode64 \the\catcode64\relax
53     \catcode123 \the\catcode123\relax
54     \catcode125 \the\catcode125\relax
55 }%
56 }%
57 \x
58 \catcode35 6 % #
59 \catcode64 11 % @
60 \catcode123 1 % {
61 \catcode125 2 % }
62 \def\TMP@EnsureCode#1#2{%
63     \edef\ACCSUPP@AtEnd{%
64         \ACCSUPP@AtEnd
65         \catcode#1 \the\catcode#1\relax
66     }%
67     \catcode#1 #2\relax
68 }
69 \TMP@EnsureCode{10}{12}% ^^J
70 \TMP@EnsureCode{33}{12}% !
71 \TMP@EnsureCode{39}{12}% '
72 \TMP@EnsureCode{40}{12}% (
73 \TMP@EnsureCode{41}{12}% )
74 \TMP@EnsureCode{42}{12}% *
75 \TMP@EnsureCode{44}{12}% ,
76 \TMP@EnsureCode{45}{12}% -
77 \TMP@EnsureCode{46}{12}% .
78 \TMP@EnsureCode{47}{12}% /
79 \TMP@EnsureCode{58}{12}% :
80 \TMP@EnsureCode{60}{12}% <
81 \TMP@EnsureCode{61}{12}% =
82 \TMP@EnsureCode{62}{12}% >
83 \TMP@EnsureCode{94}{7}% ^ (superscript)
84 \TMP@EnsureCode{96}{12}% '
85 \TMP@EnsureCode{254}{12}% ^^fe
86 \TMP@EnsureCode{255}{12}% ^^ff
87 \g@addto@macro\ACCSUPP@AtEnd{\endinput}

Package identification.
88 \NeedsTeXFormat{LaTeX2e}
89 \ProvidesPackage{accsupp}%

```

```

90 [2010/01/16 v0.3 Accessibility support by marked content (H0)]
91 \RequirePackage{pdfescape}[2007/02/25]
92 \RequirePackage{ifpdf}
93 \RequirePackage{ifxetex}
94 \RequirePackage{kvoptions}

95 \SetupKeyvalOptions{%
96   family=ACCSUPP,%
97   prefix=ACCSUPP%
98 }

```

3.2 Space setup

```

99 \DeclareBoolOption{space}
100 \DeclareStringOption[phvr8r]{spacefont}
101 \DeclareStringOption[32]{spacechar}

```

3.3 Driver detection and setup

Driver declarations.

```

102 \def\ACCSUPP@DefineDriverKey{%
103   \@dblarg\ACCSUPP@@DefineDriverKey
104 }
105 \def\ACCSUPP@@DefineDriverKey[#1]#2{%
106   \define@key{ACCSUPP}{#2}[]{%
107     \def\ACCSUPP@driver{#1}%
108   }%
109   \g@addto@macro\ACCSUPP@DisableOptions{%
110     \DisableKeyvalOption{ACCSUPP}{#2}%
111   }%
112 }
113 \let\ACCSUPP@DisableOptions\@empty
114 \ACCSUPP@DefineDriverKey{pdftex}
115 \ACCSUPP@DefineDriverKey{dvips}
116 \ACCSUPP@DefineDriverKey{dvips}{dvipsone}
117 \ACCSUPP@DefineDriverKey{dvipdfm}
118 \ACCSUPP@DefineDriverKey{dvipdfm}{dvipdfmx}
119 \ACCSUPP@DefineDriverKey{dvipdfm}{xetex}
120 \let\ACCSUPP@driver\relax
121 \InputIfFileExists{accsupp.cfg}{}{}
122 \providecommand*{\ActualTextDriverDefault}{dvips}
123 \ifpdf
124   \def\ACCSUPP@driver{pdftex}%
125 \else
126   \ifxetex
127     \def\ACCSUPP@driver{dvipdfm}%
128   \else
129     \ifx\ACCSUPP@driver\relax
130       \let\ACCSUPP@driver\ActualTextDriverDefault
131     \fi
132   \fi
133 \fi

```

Process options.

```

134 \ProcessKeyvalOptions*
135 \ACCSUPP@DisableOptions

```

Driver validation and loading.

```

136 \def\ACCSUPP@temp{pdftex}%
137 \ifpdf
138   \ifx\ACCSUPP@temp\ACCSUPP@driver
139   \else
140     \PackageWarningNoLine{accsupp}{%

```

```

141     Wrong driver '\ACCSUPP@driver', using 'pdftex' instead%
142 }%
143 \let\ACCSUPP@driver\ACCSUPP@temp
144 \fi
145 \else
146 \ifx\ACCSUPP@temp\ACCSUPP@driver
147 \PackageError{accsupp}{%
148     Wrong driver, pdfTeX is not running in PDF mode.\MessageBreak
149     Package loading is aborted%
150 }\@ehc
151 \expandafter\expandafter\expandafter\ACCSUPP@AtEnd
152 \fi
153 \def\ACCSUPP@temp{dvipdfm}%
154 \ifxetex
155 \ifx\ACCSUPP@temp\ACCSUPP@driver
156 \else
157 \PackageWarningNoLine{accsupp}{%
158     Wrong driver '\ACCSUPP@driver',\MessageBreak
159     using 'dvipdfm' for XeTeX instead%
160 }%
161 \let\ACCSUPP@driver\ACCSUPP@temp
162 \fi
163 \fi
164 \fi
165 \ifx\ACCSUPP@driver\relax
166 \PackageError{accsupp}{%
167     Missing driver option.\MessageBreak
168     Package loading is aborted%
169 }\@ehc
170 \expandafter\ACCSUPP@AtEnd
171 \fi
172 \InputIfFileExists{accsupp-\ACCSUPP@driver.def}{-}{%
173 \PackageError{accsupp}{%
174     Unsupported driver '\ACCSUPP@driver'.\MessageBreak
175     Package loading is aborted%
176 }\@ehc
177 \ACCSUPP@AtEnd
178 }

```

3.4 Main macro

```

179 \DeclareBoolOption{unicode}
180 \DeclareStringOption[page]{pdfliteral}
181 \DeclareStringOption{Lang}
182 \def\ACCSUPP@method{escape}
183 \define@key{ACCSUPP}{method}{%
184 \@ifundefined{ACCSUPP@method@#1}{%
185 \PackageError{accsupp}{%
186     Ignoring unknown method '#1'%
187 }\@ehc
188 }{%
189 \edef\ACCSUPP@method{#1}%
190 }%
191 }
192 \let\ACCSUPP@Lang\relax
193 \def\ACCSUPP@temp#1{%
194 \expandafter\ACCSUPP@@temp\csname ACCSUPP@#1\endcsname{#1}%
195 }
196 \def\ACCSUPP@@temp#1#2{%
197 \let#1\relax
198 \define@key{ACCSUPP}{#2}{%
199 \def#1{##1}%

```

```

200 \ifx#1\@empty
201 \def#1{()}%
202 \else
203 \csname ACCSUPP@method@ACCSUPP@method\endcsname#1%
204 \fi
205 }%
206 }
207 \ACCSUPP@temp{Alt}
208 \ACCSUPP@temp{ActualText}
209 \ACCSUPP@temp{E}

210 \newcommand*{\BeginAccSupp}[1]{%
211 \begingroup
212 \setkeys{ACCSUPP}{#1}%
213 \edef\ACCSUPP@span{%
214 /Span<<%
215 \ifx\ACCSUPP@Lang\relax
216 \else
217 /Lang\ACCSUPP@Lang
218 \fi
219 \ifx\ACCSUPP@Alt\relax
220 \else
221 /Alt\ACCSUPP@Alt
222 \fi
223 \ifx\ACCSUPP@ActualText\relax
224 \else
225 /ActualText\ACCSUPP@ActualText
226 \fi
227 \ifx\ACCSUPP@E\relax
228 \else
229 /E\ACCSUPP@E
230 \fi
231 >>%
232 }%
233 \ACCSUPP@bdc
234 \ACCSUPP@space
235 \endgroup
236 }

237 \newcommand*{\EndAccSupp}[1]{%
238 \begingroup
239 \setkeys{ACCSUPP}{#1}%
240 \ACCSUPP@emc
241 \endgroup
242 }

243 \def\ACCSUPP@space{%
244 \ifACCSUPP@space
245 \begingroup
246 \@ifundefined{ACCSUPP@Font}{%
247 \global\font\ACCSUPP@Font=\ACCSUPP@spacefont\relax
248 }{}%
249 \leavevmode
250 \setbox\z@\hbox{\ACCSUPP@Font\char\ACCSUPP@spacechar}%
251 \wd\z@\z@
252 \ht\z@\z@
253 \dp\z@\z@
254 \copy\z@
255 \endgroup
256 \fi
257 }

258 \newcommand*{\AccSuppSetup}{%
259 \setkeys{ACCSUPP}%
260 }

```


3.4.1 Input methods

```
261 \def\ACCSUPP@method@plain#1{%
262   \csname @safe@activestruel\endcsname
263   \edef#1{%
264     (%
265     \ifACCSUPP@unicode
266       \string\376\string\377%
267     \fi
268     #1%
269   )%
270 }%
271 \@onelevel@sanitize#1%
272 }

273 \def\ACCSUPP@method@escape#1{%
274   \EdefEscapeString#1{%
275     \ifACCSUPP@unicode
276       ^^fe^^ff%
277     \fi
278     #1%
279   }%
280   \edef#1{(#1)}%
281 }%

282 \def\ACCSUPP@method@hex#1{%
283   \edef#1{%
284     <%
285     \ifACCSUPP@unicode
286       FFFF%
287     \fi
288     #1%
289     >%
290   }%
291 }

292 \def\ACCSUPP@method@pdfstringdef#1{%
293   \ifACCSUPP@unicode
294     \@ifundefined{hypersetup}{-}{%
295       \hypersetup{unicode}%
296     }%
297   \fi
298   \@ifundefined{pdfstringdef}{%
299     \PackageError{accsupp}{%
300       Method 'pdfstringdef' requires package 'hyperref'%
301     }\@ehc
302     \let\ACCSUPP@temp\@empty
303   }{%
304     \begingroup
305       \setbox0=\hbox{%
306         \pdfstringdef\ACCSUPP@temp#1%
307         \global\let\ACCSUPP@temp\ACCSUPP@temp
308       }%
309     \endgroup
310   }%
311   \edef#1{(\ACCSUPP@temp)}%
312 }

313 \ACCSUPP@AtEnd
314 \endpackage
```

3.5 Drivers

3.5.1 Driver pdftex

```
315 \ifpdf
```

```

316 \NeedsTeXFormat{LaTeX2e}
317 \ProvidesFile{accsupp-pdftex.def}%
318 [2010/01/16 v0.3 accsupp driver for pdfTeX (H0)]%
319 \def\ACCSUPP@bdc{%
320   \pdfliteral\ACCSUPP@pdfliteral{\ACCSUPP@span BDC}%
321 }
322 \def\ACCSUPP@emc{%
323   \pdfliteral\ACCSUPP@pdfliteral{EMC}%
324 }
325 </pdftex>

```

3.5.2 Driver dvipdfm

```

326 <*dvipdfm>
327 \NeedsTeXFormat{LaTeX2e}
328 \ProvidesFile{accsupp-dvipdfm.def}%
329 [2010/01/16 v0.3 accsupp driver for dvipdfm (H0)]%
330 \def\ACCSUPP@bdc{%
331   \special{pdf:content \ACCSUPP@span BDC}%
332 }
333 \def\ACCSUPP@emc{%
334   \special{pdf:content EMC}%
335 }
336 </dvipdfm>

```

3.5.3 Driver dvips

```

337 <*dvips>
338 \NeedsTeXFormat{LaTeX2e}
339 \ProvidesFile{accsupp-dvips.def}%
340 [2010/01/16 v0.3 accsupp driver for dvips (H0)]%
341 \def\ACCSUPP@bdc{%
342   \special{ps:[\ACCSUPP@span/BDC pdfmark}%
343 }
344 \def\ACCSUPP@emc{%
345   \special{ps:[/EMC pdfmark}%
346 }
347 </dvips>

```

4 Test

4.1 Catcode checks for loading

```

348 <*test1>
349 \NeedsTeXFormat{LaTeX2e}
350 \documentclass{minimal}
351 \makeatletter
352 \def\RestoreCatcodes{%
353   \count@=0 %
354   \loop
355     \edef\RestoreCatcodes{%
356       \RestoreCatcodes
357       \catcode\the\count@=\the\catcode\count@\relax
358     }%
359   \ifnum\count@<255 %
360     \advance\count@\@ne
361   \repeat
362
363 \def\RangeCatcodeInvalid#1#2{%
364   \count@=#1\relax
365   \loop
366     \catcode\count@=15 %

```

```

367 \ifnum\count@<#2\relax
368   \advance\count@\@ne
369 \repeat
370 }
371 \def\Test{%
372   \RangeCatcodeInvalid{0}{47}%
373   \RangeCatcodeInvalid{58}{64}%
374   \RangeCatcodeInvalid{91}{96}%
375   \RangeCatcodeInvalid{123}{127}%
376   \catcode'\@=12 %
377   \catcode'\=0 %
378   \catcode'\{=1 %
379   \catcode'\}=2 %
380   \catcode'\#=6 %
381   \catcode'\[=12 %
382   \catcode'\]=12 %
383   \catcode'\%=14 %
384   \catcode'\ =10 %
385   \catcode\l3=5 %
386   \RequirePackage{accsupp}[2010/01/16]\relax
387   \RestoreCatcodes
388 }
389 \Test
390 \csname @@end\endcsname
391 \end
392 </test1>

```

5 Installation

5.1 Download

Package. This package is available on CTAN¹:

[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/
```

¹<http://ftp.ctan.org/tex-archive/>

5.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain $\mathrm{T}_{\mathrm{E}}\mathrm{X}$:

```
tex accsupp.dtx
```

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

<code>accsupp.sty</code>	\rightarrow <code>tex/latex/oberdiek/accsupp.sty</code>
<code>accsupp-pdftex.def</code>	\rightarrow <code>tex/latex/oberdiek/accsupp-pdftex.def</code>
<code>accsupp-dvipdfm.def</code>	\rightarrow <code>tex/latex/oberdiek/accsupp-dvipdfm.def</code>
<code>accsupp-dvips.def</code>	\rightarrow <code>tex/latex/oberdiek/accsupp-dvips.def</code>
<code>accsupp.pdf</code>	\rightarrow <code>doc/latex/oberdiek/accsupp.pdf</code>
<code>accsupp-example1.tex</code>	\rightarrow <code>doc/latex/oberdiek/accsupp-example1.tex</code>
<code>accsupp-example2.tex</code>	\rightarrow <code>doc/latex/oberdiek/accsupp-example2.tex</code>
<code>test/accsupp-test1.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/accsupp-test1.tex</code>
<code>accsupp.dtx</code>	\rightarrow <code>source/latex/oberdiek/accsupp.dtx</code>

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

5.4 Refresh file name databases

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

5.5 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 accsupp.pdf unpack_files output .
```

Unpacking with $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$. The `.dtx` chooses its action depending on the format:

plain $\mathrm{T}_{\mathrm{E}}\mathrm{X}$: Run `docstrip` and extract the files.

$\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$: Generate the documentation.

If you insist on using $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ for `docstrip` (really, `docstrip` does not need $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$), 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 `pdf $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$` :

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

6 References

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

7 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.

8 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		\ACCSUPP@DisableOptions	109, 113, 135
\#	380	\ACCSUPP@driver	107, 120, 124,
\%	383		127, 129, 130, 138, 141, 143,
\@	376		146, 155, 158, 161, 165, 172, 174
\@dblarg	103	\ACCSUPP@E	227, 229
\@ehc	150, 169, 176, 187, 301	\ACCSUPP@emc	240, 322, 333, 344
\@empty	113, 200, 302	\ACCSUPP@Font	247, 250
\@ifundefined	184, 246, 294, 298	\ACCSUPP@Lang	192, 215, 217
\@ne	360, 368	\ACCSUPP@method	182, 189, 203
\@onelevel@sanitize	271	\ACCSUPP@method@escape	273
\[381	\ACCSUPP@method@hex	282
\	377	\ACCSUPP@method@pdfstringdef	292
\{	378	\ACCSUPP@method@plain	261
\}	379	\ACCSUPP@pdfliteral	320, 323
\]	382	\ACCSUPP@space	234, 243
Numbers		\ACCSUPP@spacechar	250
\3	266	\ACCSUPP@spacefont	247
_	384	\ACCSUPP@span	213, 320, 331, 342
A		\ACCSUPP@temp	136, 138,
\ACCSUPP@@DefineDriverKey	103, 105		143, 146, 153, 155, 161, 193,
\ACCSUPP@temp	194, 196		207, 208, 209, 302, 306, 307, 311
\ACCSUPP@ActualText	223, 225	\AccSuppSetup	258
\ACCSUPP@Alt	219, 221	\ActualTextDriverDefault	122, 130
\ACCSUPP@AtEnd	63, 64, 87, 151, 170, 177, 313	\advance	360, 368
\ACCSUPP@bdc	233, 319, 330, 341	B	
\ACCSUPP@DefineDriverKey	102, 114, 115, 116, 117, 118, 119	\begin	17, 18, 29, 30
		\BeginAccSupp	2, 11, 31, 210
		C	
		\catcode	47,
			48, 51, 52, 53, 54, 58, 59, 60, 61,

65, 67, 357, 366, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385	\mathrel 7, 10
\centernot 12	\MessageBreak 148, 158, 167, 174
\char 250	N
\copy 254	\NeedsTeXFormat 88, 316, 327, 338, 349
\count@ 353, 357, 359, 360, 364, 366, 367, 368	\newcommand 8, 210, 237, 258
\csname 50, 194, 203, 262, 390	\notparallel 8, 19
D	P
\DeclareBoolOption 99, 179	\PackageError . 147, 166, 173, 185, 299
\DeclareStringOption 100, 101, 180, 181	\PackageWarningNoLine 140, 157
\define@key 106, 183, 198	\parallel 12
\DisableKeyvalOption 110	\pdfliteral 320, 323
\documentclass 3, 26, 350	\pdfstringdef 306
\dp 253	\ProcessKeyvalOptions 134
E	\providecommand 122
\EdefEscapeString 274	\ProvidesFile 317, 328, 339
\end 20, 21, 41, 42, 391	\ProvidesPackage 89
\EndAccSupp 2, 13, 40, 237	R
\endcsname 50, 194, 203, 262, 390	\RangeCatcodeInvalid 363, 372, 373, 374, 375
\endinput 87	\repeat 361, 369
\ensuremath 9	\RequirePackage . . . 91, 92, 93, 94, 386
F	\RestoreCatcodes . . 352, 355, 356, 387
\font 247	S
G	\setbox 250, 305
\g@addto@macro 87, 109	\setkeys 212, 239, 259
H	\SetupKeyvalOptions 95
\hbox 250, 305	\special 331, 334, 342, 345
\ht 252	T
\hypersetup 295	\Test 371, 389
I	\texttwosuperior 35, 36
\ifACCSUPP@space 244	\the 51, 52, 53, 54, 65, 357
\ifACCSUPP@unicode . 265, 275, 285, 293	\TMP@EnsureCode 62, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86
\ifnum 359, 367	U
\ifpdf 123, 137	\usepackage 4, 5, 27, 28
\ifx 129, 138, 146, 155, 165, 200, 215, 219, 223, 227	W
\ifxetex 126, 154	\wd 251
\InputIfFileExists 121, 172	X
L	\x 49, 57
\leavevmode 249	Z
\loop 354, 365	\z@ 250, 251, 252, 253, 254
M	
\makeatletter 351	