

The `accsupp` package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/03/21 v0.1

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	Driver options	3
1.2.1	Option <code>pdftex</code>	3
1.2.2	Option <code>dvipdfm</code>	3
1.2.3	Option <code>dvips</code>	3
1.2.4	Turning off page stream compression	3
2	Example	3
3	Implementation	4
3.1	Package	4
3.2	Driver detection and setup	4
3.3	Main macro	5
3.3.1	Input methods	7
3.4	Drivers	7
3.4.1	Driver <code>pdftex</code>	7
3.4.2	Driver <code>dvipdfm</code>	8
3.4.3	Driver <code>dvips</code>	8
4	Installation	8
4.1	Download	8
4.2	Bundle installation	8
4.3	Package installation	9
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	References	10
6	History	10
	[2007/03/21 v0.1]	10
7	Index	10

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 1.2.1.

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 example 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 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 Driver options

Driver options are package options only. The special \TeX compilers $\text{pdf}\text{\TeX}$ and $\text{Xe}\text{\TeX}$ are detected automatically. The default for unrecognized drivers can be set by defining `\ActualTextDriverDefault`. This can be done in the configuration file `accsupp.cfg`.

1.2.1 Option `pdftex`

Package option `pdftex` is used for $\text{pdf}\text{\TeX}$ in PDF mode. Additionally `\BeginAccSupp` and `\EndAccSupp` understand option `pdfliteral`. It controls the modifier keyword for `\pdfliteral`:

```
pdfliteral=direct  $\Rightarrow$  \pdfliteral direct{...}
```

1.2.2 Option `dvipdfm`

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

1.2.3 Option `dvips`

Package option `dvips` and its alias `dvipson` 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.

1.2.4 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:

```
 $\text{pdf}\text{\TeX}$ : \pdfcompresslevel=0
```

```
dvipdfm: dvipdfm -z0 ...
```

```
dvipdfmx: dvipdfmx -z0 ...
```

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

2 Example

```
1 \(*example)
2 \documentclass{article}
3 \usepackage{accsupp}
4 \usepackage{centernot}
5 % U+2226 NOT PARALLEL
6 % \mathrel{...} prevents page break in between
7 \newcommand*{\notparallel}{%
8   \ensuremath{%
9     \mathrel{%
10       \BeginAccSupp[method=hex,unicode,ActualText=2226]%
11       \centernot{\parallel}%
12       \EndAccSupp{}}%
13   }%
14 }%
15 }
16 \begin{document}
17 \begin{equation}
```

```

18 A\notparallel B
19 \end{equation}
20 \end{document}
21 \end{example}

```

3 Implementation

3.1 Package

```

22 (*package)
23 \NeedsTeXFormat{LaTeX2e}
24 \ProvidesPackage{accsupp}%
25   [2007/03/21 v0.1 Accessibility support by marked content (HO)]%
26 \edef\ACCSUPP@end{%
27   \catcode39 \the\catcode39 % '
28   \catcode40 \the\catcode40 % (
29   \catcode41 \the\catcode41 % )
30   \catcode60 \the\catcode60 % <
31   \catcode61 \the\catcode61 % =
32   \catcode62 \the\catcode62 % >
33   \catcode94 \the\catcode94 % ^
34   \catcode96 \the\catcode96 % '
35   \catcode254 \the\catcode254 % ^^FE
36   \catcode255 \the\catcode255 % ^^FF
37 }
38 \@makeother\<
39 \@makeother\>
40 \@makeother\'
41 \@makeother\<
42 \@makeother\=
43 \@makeother\>
44 \catcode94 = 7 %
45 \@makeother\'
46 \@makeother^^fe
47 \@makeother^^ff
48 \RequirePackage{pdfescape}[2007/02/25]
49 \RequirePackage{ifpdf}
50 \RequirePackage{ifxetex}
51 \RequirePackage{kvoptions}
52 \SetupKeyvalOptions{%
53   family=ACCSUPP,%
54   prefix=ACCSUPP%
55 }

```

3.2 Driver detection and setup

Driver declarations.

```

56 \def\ACCSUPP@DefineDriverKey{%
57   \@dblarg\ACCSUPP@@DefineDriverKey
58 }
59 \def\ACCSUPP@@DefineDriverKey[#1]#2{%
60   \define@key{ActTxt}{#2}[]{%
61     \def\ACCSUPP@driver{#1}%
62   }%
63   \g@addto@macro\ACCSUPP@DisableOptions{%
64     \DisableKeyvalOption{ActTxt}{#2}%
65   }%
66 }
67 \let\ACCSUPP@DisableOptions\@empty
68 \ACCSUPP@DefineDriverKey{pdftex}
69 \ACCSUPP@DefineDriverKey{dvips}
70 \ACCSUPP@DefineDriverKey{dvips}[dvipsone]

```

```

71 \ACCSUPP@DefineDriverKey{dvipdfm}
72 \ACCSUPP@DefineDriverKey[dvipdfm]{dvipdfmx}
73 \ACCSUPP@DefineDriverKey[dvipdfm]{xetex}
74 \InputIfFileExists{accsupp.cfg}{-}{-}
75 \providecommand*{\ActualTextDriverDefault}{dvips}
76 \ifpdf
77   \def\ACCSUPP@driver{pdftex}%
78 \else
79   \ifxetex
80     \def\ACCSUPP@driver{dvipdfm}%
81   \else
82     \let\ACCSUPP@driver\BookmarkDriverDefault
83   \fi
84 \fi

```

Process options.

```

85 \ProcessKeyvalOptions*
86 \ACCSUPP@DisableOptions

```

Driver validation and loading.

```

87 \def\ACCSUPP@temp{pdftex}%
88 \ifpdf
89   \ifx\ACCSUPP@temp\ACCSUPP@driver
90   \else
91     \PackageWarningNoLine{accsupp}{%
92       Wrong driver '\ACCSUPP@driver', using 'pdftex' instead%
93     }%
94     \let\ACCSUPP@driver\ACCSUPP@temp
95   \fi
96 \else
97   \ifx\ACCSUPP@temp\ACCSUPP@driver
98     \PackageError{accsupp}{%
99       Wrong driver, pdfTeX is not running in PDF mode.\MessageBreak
100       Package loading is aborted%
101     }\@ehc
102     \expandafter\expandafter\expandafter\endinput
103   \fi
104   \def\ACCSUPP@temp{dvipdfm}%
105   \ifxetex
106     \ifx\ACCSUPP@temp\ACCSUPP@driver
107     \else
108       \PackageWarningNoLine{accsupp}{%
109         Wrong driver '\ACCSUPP@driver',\MessageBreak
110         using 'dvipdfm' for XeTeX instead%
111       }%
112       \let\ACCSUPP@driver\ACCSUPP@temp
113     \fi
114   \fi
115 \fi
116 \InputIfFileExists{accsupp-\ACCSUPP@driver.def}{-}{-}%
117 \PackageError{accsupp}{%
118   Unsupported driver '\ACCSUPP@driver'.\MessageBreak
119   Package loading is aborted%
120 }\@ehc
121 \endinput
122 }

```

3.3 Main macro

```

123 \DeclareBoolOption{unicode}
124 \DeclareStringOption[page]{pdfliteral}

```

```

125 \DeclareStringOption{Lang}
126 \def\ACCSUPP@method{escape}
127 \define@key{ACCSUPP}{method}{%
128   \@ifundefined{ACCSUPP@method@#1}{%
129     \PackageError{accsupp}{%
130       Ignoring unknown method ‘#1’%
131     }\@ehc
132   }{%
133     \edef\ACCSUPP@method{#1}%
134   }%
135 }
136 \let\ACCSUPP@Lang\relax
137 \def\ACCSUPP@temp#1{%
138   \expandafter\ACCSUPP@@temp\csname ACCSUPP@#1\endcsname{#1}%
139 }
140 \def\ACCSUPP@@temp#1#2{%
141   \let#1\relax
142   \define@key{ACCSUPP}{#2}{%
143     \def#1{##1}%
144     \ifx#1\@empty
145       \def#1{()%
146     \else
147       \csname ACCSUPP@method@\ACCSUPP@method\endcsname#1%
148     \fi
149   }%
150 }
151 \ACCSUPP@temp{Alt}
152 \ACCSUPP@temp{ActualText}
153 \ACCSUPP@temp{E}
154 \newcommand*\BeginAccSupp[1]{%
155   \begingroup
156     \setkeys{ACCSUPP}{#1}%
157     \edef\ACCSUPP@span{%
158       /Span<<%
159       \ifx\ACCSUPP@Lang\relax
160         \else
161           /Lang\ACCSUPP@Lang
162         \fi
163       \ifx\ACCSUPP@Alt\relax
164         \else
165           /Alt\ACCSUPP@Alt
166         \fi
167       \ifx\ACCSUPP@ActualText\relax
168         \else
169           /ActualText\ACCSUPP@ActualText
170         \fi
171       \ifx\ACCSUPP@E\relax
172         \else
173           /E\ACCSUPP@E
174         \fi
175       >>%
176     }%
177     \ACCSUPP@bdc
178   \endgroup
179 }
180 \newcommand*\EndAccSupp[1]{%
181   \begingroup
182     \setkeys{ACCSUPP}{#1}%
183     \ACCSUPP@emc
184   \endgroup
185 }

```

3.3.1 Input methods

```
186 \def\ACCSUPP@method@plain#1{%
187   \csname @safe@activetrue\endcsname
188   \edef#1{%
189     (%
190     \ifACCSUPP@unicode
191       \string\376\string\377%
192     \fi
193     #1%
194     )%
195   }%
196   \@onelevel@sanitize#1%
197 }

198 \def\ACCSUPP@method@escape#1{%
199   \EdefEscapeString#1{%
200     \ifACCSUPP@unicode
201       ^^fe^^ff%
202     \fi
203     #1%
204   }%
205   \edef#1{(#1)}%
206 }%

207 \def\ACCSUPP@method@hex#1{%
208   \edef#1{%
209     <%
210     \ifACCSUPP@unicode
211       FFFF%
212     \fi
213     #1%
214     >%
215   }%
216 }

217 \def\ACCSUPP@method@pdfstringdef#1{%
218   \ifActText@unicode
219     \@ifundefined{hypersetup}{-}{%
220       \hypersetup{unicode}%
221     }%
222   \fi
223   \@ifundefined{pdfstringdef}{%
224     \PackageError{accsupp}{%
225       Method 'pdfstringdef' requires package 'hyperref'%
226     }{\@ehc}
227   \let\ACCSUPP@temp\@empty
228   }{%
229     \pdfstringdef\ACCSUPP@temp#1%
230   }%
231   \edef#1{(\ACCSUPP@temp)}%
232 }

233 \ACCSUPP@end
234 \end{package}
```

3.4 Drivers

3.4.1 Driver pdftex

```
235 (*pdftex)
236 \NeedsTeXFormat{LaTeX2e}
237 \ProvidesFile{accsupp-pdftex.def}%
238   [2007/03/21 v0.1 accsupp driver for pdfTeX (H0)]%

239 \def\ACCSUPP@bdc{%
240   \pdfliteral\ACCSUPP@pdfliteral{\ACCSUPP@span BDC}%

```

```

241 }
242 \def\ACCSUPP@emc{%
243   \pdfliteral\ACCSUPP@pdfliteral{EMC}%
244 }
245 </pdfTeX>

3.4.2 Driver dvipdfm

246 <*dvipdfm>
247 \NeedsTeXFormat{LaTeX2e}
248 \ProvidesFile{accsupp-dvipdfm.def}%
249   [2007/03/21 v0.1 accsupp driver for dvipdfm (HO)]%

250 \def\ACCSUPP@bdc{%
251   \special{pdf:content \ACCSUPP@span BDC}%
252 }
253 \def\ACCSUPP@emc{%
254   \special{pdf:content EMC}%
255 }

256 </dvipdfm>

3.4.3 Driver dvips

257 <*dvips>
258 \NeedsTeXFormat{LaTeX2e}
259 \ProvidesFile{accsupp-dvips.def}%
260   [2007/03/21 v0.1 accsupp driver for dvips (HO)]%

261 \def\ACCSUPP@bdc#1{%
262   \special{ps:[\ACCSUPP@span/BDC pdfmark}%
263 }
264 \def\ACCSUPP@emc{%
265   \special{ps:[/EMC pdfmark}%
266 }

267 </dvips>

```

4 Installation

4.1 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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

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
```

¹<http://ftp.ctan.org/tex-archive/>

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 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>	→	<code>tex/latex/oberdiek/accsupp.sty</code>
<code>accsupp-pdftex.def</code>	→	<code>tex/latex/oberdiek/accsupp-pdftex.def</code>
<code>accsupp-dvipdfm.def</code>	→	<code>tex/latex/oberdiek/accsupp-dvipdfm.def</code>
<code>accsupp-dvips.def</code>	→	<code>tex/latex/oberdiek/accsupp-dvips.def</code>
<code>accsupp.pdf</code>	→	<code>doc/latex/oberdiek/accsupp.pdf</code>
<code>accsupp-example.tex</code>	→	<code>doc/latex/oberdiek/accsupp-example.tex</code>
<code>accsupp.dtx</code>	→	<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`.

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 `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk accsupp.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{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 pdfL^AT_EX:

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

5 References

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

6 History

[2007/03/21 v0.1]

- First version.

7 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		
\'	40	\ACCSUPP@DisableOptions .. 63, 67, 86
\(38	\ACCSUPP@driver 61, 77, 80, 82, 89,
\)	39	92, 94, 97, 106, 109, 112, 116, 118
\<	41	\ACCSUPP@E .. 171, 173
\=	42	\ACCSUPP@emc .. 183, 242, 253, 264
\>	43	\ACCSUPP@end .. 26, 233
\@dblarg	57	\ACCSUPP@Lang .. 136, 159, 161
\@ehc	101, 120, 131, 226	\ACCSUPP@method .. 126, 133, 147
\@empty	67, 144, 227	\ACCSUPP@method@escape .. 198
\@ifundefined	128, 219, 223	\ACCSUPP@method@hex .. 207
\@makeother	38, 39, 40, 41, 42, 43, 45, 46, 47	\ACCSUPP@method@pdfstringdef .. 217
\@onelevel@sanitize	196	\ACCSUPP@method@plain .. 186
\^	46, 47	\ACCSUPP@pdfliteral .. 240, 243
\'	45	\ACCSUPP@span .. 157, 240, 251, 262
		\ACCSUPP@temp .. 87, 89, 94, 97, 104, 106, 112,
		137, 151, 152, 153, 227, 229, 231
Numbers		\ActualTextDriverDefault .. 75
\3	191	
A		B
\ACCSUPP@@DefineDriverKey	57, 59	\begin .. 16, 17
\ACCSUPP@@temp	138, 140	\BeginAccSupp .. 2, 10, 154
\ACCSUPP@ActualText	167, 169	\BookmarkDriverDefault .. 82
\ACCSUPP@Alt	163, 165	C
\ACCSUPP@bdc	177, 239, 250, 261	\catcode .. 27,
\ACCSUPP@DefineDriverKey	56, 68, 69, 70, 71, 72, 73	28, 29, 30, 31, 32, 33, 34, 35, 36, 44
		\centernot .. 11

<code>\csname</code>	138, 147, 187	<code>\MessageBreak</code>	99, 109, 118
D		N	
<code>\DeclareBoolOption</code>	123	<code>\NeedsTeXFormat</code>	23, 236, 247, 258
<code>\DeclareStringOption</code>	124, 125	<code>\newcommand</code>	7, 154, 180
<code>\define@key</code>	60, 127, 142	<code>\notparallel</code>	7, 18
<code>\DisableKeyvalOption</code>	64	P	
<code>\documentclass</code>	2	<code>\PackageError</code>	98, 117, 129, 224
E		<code>\PackageWarningNoLine</code>	91, 108
<code>\EdefEscapeString</code>	199	<code>\parallel</code>	11
<code>\end</code>	19, 20	<code>\pdfliteral</code>	240, 243
<code>\EndAccSupp</code>	2, 12, 180	<code>\pdfstringdef</code>	229
<code>\endcsname</code>	138, 147, 187	<code>\ProcessKeyvalOptions</code>	85
<code>\endinput</code>	102, 121	<code>\providecommand</code>	75
<code>\ensuremath</code>	8	<code>\ProvidesFile</code>	237, 248, 259
G		<code>\ProvidesPackage</code>	24
<code>\g@addto@macro</code>	63	R	
H		<code>\RequirePackage</code>	48, 49, 50, 51
<code>\hypersetup</code>	220	S	
I		<code>\setkeys</code>	156, 182
<code>\ifACCSUPP@unicode</code>	190, 200, 210	<code>\SetupKeyvalOptions</code>	52
<code>\ifActText@unicode</code>	218	<code>\special</code>	251, 254, 262, 265
<code>\ifpdf</code>	76, 88	T	
<code>\ifx</code> . 89, 97, 106, 144, 159, 163, 167, 171		<code>\the</code> . 27, 28, 29, 30, 31, 32, 33, 34, 35, 36	
<code>\ifxetex</code>	79, 105	U	
<code>\InputIfFileExists</code>	74, 116	<code>\usepackage</code>	3, 4
M			
<code>\mathrel</code>	6, 9		