

The `accsupp` package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/11/14 v0.2

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 |
| 2.1 | Example <code>\notparallel</code> | 3 |
| 2.2 | Example with <code>pdfstringdef</code> | 4 |
| 3 | Implementation | 4 |
| 3.1 | Package | 4 |
| 3.2 | Driver detection and setup | 5 |
| 3.3 | Main macro | 7 |
| 3.3.1 | Input methods | 8 |
| 3.4 | Drivers | 9 |
| 3.4.1 | Driver <code>pdftex</code> | 9 |
| 3.4.2 | Driver <code>dvipdfm</code> | 9 |
| 3.4.3 | Driver <code>dvips</code> | 9 |
| 4 | Test | 9 |
| 4.1 | Catcode checks for loading | 9 |
| 5 | Installation | 10 |
| 5.1 | Download | 10 |
| 5.2 | Bundle installation | 10 |
| 5.3 | Package installation | 11 |
| 5.4 | Refresh file name databases | 11 |
| 5.5 | Some details for the interested | 11 |
| 6 | References | 12 |

| | |
|-----------------------------|-----------|
| 7 History | 12 |
| [2007/03/21 v0.1] | 12 |
| [2007/11/14 v0.2] | 12 |
| 8 Index | 12 |

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 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 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`.

1.2.1 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  $\Rightarrow$  \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 `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.

1.2.4 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:

```
pdf $\TeX$ : \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\texttt{two}superior +b\texttt{two}superior
36         =c\texttt{two}superior
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
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   [2007/11/14 v0.2 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 Driver detection and setup

Driver declarations.

```

99 \def\ACCSUPP@DefineDriverKey{%
100   \@dblarg\ACCSUPP@@DefineDriverKey
101 }
102 \def\ACCSUPP@@DefineDriverKey[#1]#2{%
103   \define@key{ACCSUPP}{#2}[]{%
104     \def\ACCSUPP@driver{#1}%
105   }%
106   \g@addto@macro\ACCSUPP@DisableOptions{%
107     \DisableKeyvalOption{ACCSUPP}{#2}%
108   }%
109 }
110 \let\ACCSUPP@DisableOptions\@empty
111 \ACCSUPP@DefineDriverKey{pdftex}
112 \ACCSUPP@DefineDriverKey{dvips}
113 \ACCSUPP@DefineDriverKey{dvips}{dvipsone}
114 \ACCSUPP@DefineDriverKey{dvipdfm}
115 \ACCSUPP@DefineDriverKey{dvipdfm}{dvipdfmx}
116 \ACCSUPP@DefineDriverKey{dvipdfm}{xetex}
117 \let\ACCSUPP@driver\relax
118 \InputIfFileExists{accsupp.cfg}{}{}

```

```

119 \providecommand*{\ActualTextDriverDefault}{dvips}
120 \ifpdf
121   \def\ACCSUPP@driver{pdftex}%
122 \else
123   \ifxetex
124     \def\ACCSUPP@driver{dvipdfm}%
125   \else
126     \ifx\ACCSUPP@driver\relax
127       \let\ACCSUPP@driver\ActualTextDriverDefault
128     \fi
129   \fi
130 \fi

```

Process options.

```

131 \ProcessKeyvalOptions*
132 \ACCSUPP@DisableOptions

```

Driver validation and loading.

```

133 \def\ACCSUPP@temp{pdftex}%
134 \ifpdf
135   \ifx\ACCSUPP@temp\ACCSUPP@driver
136   \else
137     \PackageWarningNoLine{accsupp}{%
138       Wrong driver '\ACCSUPP@driver', using 'pdftex' instead%
139     }%
140     \let\ACCSUPP@driver\ACCSUPP@temp
141   \fi
142 \else
143   \ifx\ACCSUPP@temp\ACCSUPP@driver
144     \PackageError{accsupp}{%
145       Wrong driver, pdfTeX is not running in PDF mode.\MessageBreak
146       Package loading is aborted%
147     }\@ehc
148     \expandafter\expandafter\expandafter\ACCSUPP@AtEnd
149   \fi
150   \def\ACCSUPP@temp{dvipdfm}%
151   \ifxetex
152     \ifx\ACCSUPP@temp\ACCSUPP@driver
153     \else
154       \PackageWarningNoLine{accsupp}{%
155         Wrong driver '\ACCSUPP@driver',\MessageBreak
156         using 'dvipdfm' for XeTeX instead%
157       }%
158       \let\ACCSUPP@driver\ACCSUPP@temp
159     \fi
160   \fi
161 \fi
162 \ifx\ACCSUPP@driver\relax
163   \PackageError{accsupp}{%
164     Missing driver option.\MessageBreak
165     Package loading is aborted%
166   }\@ehc
167   \expandafter\ACCSUPP@AtEnd
168 \fi
169 \InputIfFileExists{accsupp-\ACCSUPP@driver.def}{}{%
170   \PackageError{accsupp}{%
171     Unsupported driver '\ACCSUPP@driver'.\MessageBreak
172     Package loading is aborted%
173   }\@ehc
174   \ACCSUPP@AtEnd
175 }

```

3.3 Main macro

```
176 \DeclareBoolOption{unicode}
177 \DeclareStringOption[page]{pdfliteral}
178 \DeclareStringOption{Lang}
179 \def\ACCSUPP@method{escape}
180 \define@key{ACCSUPP}{method}{%
181   \ifundefined{ACCSUPP@method@#1}{%
182     \PackageError{accsupp}{%
183       Ignoring unknown method ‘#1’%
184     }\@ehc
185   }{%
186     \edef\ACCSUPP@method{#1}%
187   }%
188 }
189 \let\ACCSUPP@Lang\relax
190 \def\ACCSUPP@temp#1{%
191   \expandafter\ACCSUPP@@temp\csname ACCSUPP@#1\endcsname{#1}%
192 }
193 \def\ACCSUPP@@temp#1#2{%
194   \let#1\relax
195   \define@key{ACCSUPP}{#2}{%
196     \def#1{##1}%
197     \ifx#1\@empty
198       \def#1{()}%
199     \else
200       \csname ACCSUPP@method@\ACCSUPP@method\endcsname#1%
201     \fi
202   }%
203 }
204 \ACCSUPP@temp{Alt}
205 \ACCSUPP@temp{ActualText}
206 \ACCSUPP@temp{E}
207 \newcommand*{\BeginAccSupp}[1]{%
208   \begingroup
209   \setkeys{ACCSUPP}{#1}%
210   \edef\ACCSUPP@span{%
211     /Span<%
212     \ifx\ACCSUPP@Lang\relax
213     \else
214       /Lang\ACCSUPP@Lang
215     \fi
216     \ifx\ACCSUPP@Alt\relax
217     \else
218       /Alt\ACCSUPP@Alt
219     \fi
220     \ifx\ACCSUPP@ActualText\relax
221     \else
222       /ActualText\ACCSUPP@ActualText
223     \fi
224     \ifx\ACCSUPP@E\relax
225     \else
226       /E\ACCSUPP@E
227     \fi
228     >>%
229   }%
230   \ACCSUPP@bdc
231 \endgroup
232 }
233 \newcommand*{\EndAccSupp}[1]{%
234   \begingroup
235   \setkeys{ACCSUPP}{#1}%
236   \ACCSUPP@emc
```

```

237 \endgroup
238 }

```

3.3.1 Input methods

```

239 \def\ACCSUPP@method@plain#1{%
240   \csname @safe@activestrue\endcsname
241   \edef#1{%
242     (%
243     \ifACCSUPP@unicode
244       \string\376\string\377%
245     \fi
246     #1%
247   )%
248   }%
249   \@onelevel@sanitize#1%
250 }

251 \def\ACCSUPP@method@escape#1{%
252   \EdefEscapeString#1{%
253     \ifACCSUPP@unicode
254       ^^fe^^ff%
255     \fi
256     #1%
257   }%
258   \edef#1{(#1)}%
259 }%

260 \def\ACCSUPP@method@hex#1{%
261   \edef#1{%
262     <%
263     \ifACCSUPP@unicode
264       FFFF%
265     \fi
266     #1%
267     >%
268   }%
269 }

270 \def\ACCSUPP@method@pdfstringdef#1{%
271   \ifACCSUPP@unicode
272     \@ifundefined{hypersetup}{-}{%
273       \hypersetup{unicode}%
274     }%
275   \fi
276   \@ifundefined{pdfstringdef}{-}{%
277     \PackageError{accsupp}{%
278       Method ‘pdfstringdef’ requires package ‘hyperref’%
279     }\@ehc
280     \let\ACCSUPP@temp\@empty
281   }{%
282     \begingroup
283     \setbox0=\hbox{%
284       \pdfstringdef\ACCSUPP@temp#1%
285       \global\let\ACCSUPP@temp\ACCSUPP@temp
286     }%
287     \endgroup
288   }%
289   \edef#1{(\ACCSUPP@temp)}%
290 }

291 \ACCSUPP@AtEnd
292 </package>

```


3.4 Drivers

3.4.1 Driver pdftex

```
293 <*pdftex>
294 \NeedsTeXFormat{LaTeX2e}
295 \ProvidesFile{accsupp-pdftex.def}%
296   [2007/11/14 v0.2 accsupp driver for pdfTeX (HO)]%
297 \def\ACCSUPP@bdc{%
298   \pdfliteral\ACCSUPP@pdfliteral{\ACCSUPP@span BDC}%
299 }
300 \def\ACCSUPP@emc{%
301   \pdfliteral\ACCSUPP@pdfliteral{EMC}%
302 }
303 </pdftex>
```

3.4.2 Driver dvipdfm

```
304 <*dvipdfm>
305 \NeedsTeXFormat{LaTeX2e}
306 \ProvidesFile{accsupp-dvipdfm.def}%
307   [2007/11/14 v0.2 accsupp driver for dvipdfm (HO)]%
308 \def\ACCSUPP@bdc{%
309   \special{pdf:content \ACCSUPP@span BDC}%
310 }
311 \def\ACCSUPP@emc{%
312   \special{pdf:content EMC}%
313 }
314 </dvipdfm>
```

3.4.3 Driver dvips

```
315 <*dvips>
316 \NeedsTeXFormat{LaTeX2e}
317 \ProvidesFile{accsupp-dvips.def}%
318   [2007/11/14 v0.2 accsupp driver for dvips (HO)]%
319 \def\ACCSUPP@bdc{%
320   \special{ps:[\ACCSUPP@span/BDC pdfmark}%
321 }
322 \def\ACCSUPP@emc{%
323   \special{ps:[/EMC pdfmark}%
324 }
325 </dvips>
```

4 Test

4.1 Catcode checks for loading

```
326 <*test1>
327 \NeedsTeXFormat{LaTeX2e}
328 \documentclass{minimal}
329 \makeatletter
330 \def\RestoreCatcodes{}
331 \count@=0 %
332 \loop
333   \edef\RestoreCatcodes{%
334     \RestoreCatcodes
335     \catcode\the\count@=\the\catcode\count@\relax
336   }%
337 \ifnum\count@<255 %
338   \advance\count@\@ne
339 \repeat
```

```

340
341 \def\RangeCatcodeInvalid#1#2{%
342   \count@=#1\relax
343   \loop
344     \catcode\count@=15 %
345     \ifnum\count@<#2\relax
346       \advance\count@\@ne
347     \repeat
348 }
349 \def\Test{%
350   \RangeCatcodeInvalid{0}{47}%
351   \RangeCatcodeInvalid{58}{64}%
352   \RangeCatcodeInvalid{91}{96}%
353   \RangeCatcodeInvalid{123}{127}%
354   \catcode'\@=12 %
355   \catcode'\=0 %
356   \catcode'\{=1 %
357   \catcode'\}=2 %
358   \catcode'\#=6 %
359   \catcode'\[=12 %
360   \catcode'\]=12 %
361   \catcode'\%=14 %
362   \catcode'\ =10 %
363   \catcode13=5 %
364   \RequirePackage{accsupp}[2007/11/14]\relax
365   \RestoreCatcodes
366 }
367 \Test
368 \csname @@end\endcsname
369 \end
370 </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
```

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/
```

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
```

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-example1.tex</code> | → <code>doc/latex/oberdiek/accsupp-example1.tex</code> |
| <code>accsupp-example2.tex</code> | → <code>doc/latex/oberdiek/accsupp-example2.tex</code> |
| <code>test/accsupp-test1.tex</code> | → <code>doc/latex/oberdiek/test/accsupp-test1.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.

5.4 Refresh file name databases

If your \TeX distribution (te \TeX , mik \TeX , ...) relies on file name databases, you must refresh these. For example, te \TeX 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 \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
```

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.

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 | | A | |
|---------------------|-------------------------|------------------------------|--|
| \# | 358 | \ACCSUPP@@DefineDriverKey | 100, 102 |
| \% | 361 | \ACCSUPP@temp | 191, 193 |
| \@ | 354 | \ACCSUPP@ActualText | 220, 222 |
| \@dblarg | 100 | \ACCSUPP@Alt | 216, 218 |
| \@ehc | 147, 166, 173, 184, 279 | \ACCSUPP@AtEnd | 63, 64, 87, 148, 167, 174, 291 |
| \@empty | 110, 197, 280 | \ACCSUPP@bdc | 230, 297, 308, 319 |
| \@ifundefined | 181, 272, 276 | \ACCSUPP@DefineDriverKey | 99, 111, 112, 113, 114, 115, 116 |
| \@ne | 338, 346 | \ACCSUPP@DisableOptions | 106, 110, 132 |
| \@onelevel@sanitize | 249 | \ACCSUPP@driver | 104, 117, 121, 124, 126, 127, 135, 138, 140, 143, 152, 155, 158, 162, 169, 171 |
| \[| 359 | \ACCSUPP@E | 224, 226 |
| \] | 355 | \ACCSUPP@emc | 236, 300, 311, 322 |
| \{ | 356 | \ACCSUPP@Lang | 189, 212, 214 |
| \} | 357 | \ACCSUPP@method | 179, 186, 200 |
| \] | 360 | \ACCSUPP@method@escape | 251 |
| | | \ACCSUPP@method@hex | 260 |
| | | \ACCSUPP@method@pdfstringdef | 270 |
| | | \ACCSUPP@method@plain | 239 |
| Numbers | | | |
| \3 | 244 | | |
| _ | 362 | | |

| | | | |
|--|---|--|--|
| <code>\ACCSUPP@pdfliteral</code> | 298, 301 | | |
| <code>\ACCSUPP@span</code> | 210, 298, 309, 320 | | |
| <code>\ACCSUPP@temp</code> | 133, 135, 140, 143, 150, 152, 158, 190, 204, 205, 206, 280, 284, 285, 289 | | |
| <code>\ActualTextDriverDefault</code> . . . | 119, 127 | | |
| <code>\advance</code> | 338, 346 | | |
| B | | | |
| <code>\begin</code> | 17, 18, 29, 30 | | |
| <code>\BeginAccSupp</code> | 2, 11, 31, 207 | | |
| C | | | |
| <code>\catcode</code> | 47, 48, 51, 52, 53, 54, 58, 59, 60, 61, 65, 67, 335, 344, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363 | | |
| <code>\centernot</code> | 12 | | |
| <code>\count@</code> | 331, 335, 337, 338, 342, 344, 345, 346 | | |
| <code>\csname</code> | 50, 191, 200, 240, 368 | | |
| D | | | |
| <code>\DeclareBoolOption</code> | 176 | | |
| <code>\DeclareStringOption</code> | 177, 178 | | |
| <code>\define@key</code> | 103, 180, 195 | | |
| <code>\DisableKeyvalOption</code> | 107 | | |
| <code>\documentclass</code> | 3, 26, 328 | | |
| E | | | |
| <code>\EdefEscapeString</code> | 252 | | |
| <code>\end</code> | 20, 21, 41, 42, 369 | | |
| <code>\EndAccSupp</code> | 2, 13, 40, 233 | | |
| <code>\endcsname</code> | 50, 191, 200, 240, 368 | | |
| <code>\endinput</code> | 87 | | |
| <code>\ensuremath</code> | 9 | | |
| G | | | |
| <code>\g@addto@macro</code> | 87, 106 | | |
| H | | | |
| <code>\hbox</code> | 283 | | |
| <code>\hypersetup</code> | 273 | | |
| I | | | |
| <code>\ifACCSUPP@unicode</code> | 243, 253, 263, 271 | | |
| <code>\ifnum</code> | 337, 345 | | |
| <code>\ifpdf</code> | 120, 134 | | |
| <code>\ifx</code> | 126, 135, 143, 152, 162, 197, 212, 216, 220, 224 | | |
| <code>\ifxetex</code> | 123, 151 | | |
| <code>\InputIfFileExists</code> | 118, 169 | | |
| L | | | |
| <code>\loop</code> | 332, 343 | | |
| M | | | |
| <code>\makeatletter</code> | 329 | | |
| <code>\mathrel</code> | 7, 10 | | |
| <code>\MessageBreak</code> | 145, 155, 164, 171 | | |
| N | | | |
| <code>\NeedsTeXFormat</code> | 88, 294, 305, 316, 327 | | |
| <code>\newcommand</code> | 8, 207, 233 | | |
| <code>\notparallel</code> | 8, 19 | | |
| P | | | |
| <code>\PackageError</code> | 144, 163, 170, 182, 277 | | |
| <code>\PackageWarningNoLine</code> | 137, 154 | | |
| <code>\parallel</code> | 12 | | |
| <code>\pdfliteral</code> | 298, 301 | | |
| <code>\pdfstringdef</code> | 284 | | |
| <code>\ProcessKeyvalOptions</code> | 131 | | |
| <code>\providecommand</code> | 119 | | |
| <code>\ProvidesFile</code> | 295, 306, 317 | | |
| <code>\ProvidesPackage</code> | 89 | | |
| R | | | |
| <code>\RangeCatcodeInvalid</code> | 341, 350, 351, 352, 353 | | |
| <code>\repeat</code> | 339, 347 | | |
| <code>\RequirePackage</code> | 91, 92, 93, 94, 364 | | |
| <code>\RestoreCatcodes</code> | 330, 333, 334, 365 | | |
| S | | | |
| <code>\setbox</code> | 283 | | |
| <code>\setkeys</code> | 209, 235 | | |
| <code>\SetupKeyvalOptions</code> | 95 | | |
| <code>\special</code> | 309, 312, 320, 323 | | |
| T | | | |
| <code>\Test</code> | 349, 367 | | |
| <code>\textttwosuperior</code> | 35, 36 | | |
| <code>\the</code> | 51, 52, 53, 54, 65, 335 | | |
| <code>\TMP@EnsureCode</code> | 62, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86 | | |
| U | | | |
| <code>\usepackage</code> | 4, 5, 27, 28 | | |
| X | | | |
| <code>\x</code> | 49, 57 | | |