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 space . . . 3
1.3 Driver options . . . 3
1.3.1 Option pdftex . . . 3
1.3.2 Option dvipdfm . . . 3
1.3.3 Option dvips . . . 4
1.3.4 Turning off page stream compression . . . 4

2 Example **4**
2.1 Example \notparallel . . . 4
2.2 Example with pdfstringdef . . . 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 . . . 10
3.5.1 Driver pdftex . . . 10
3.5.2 Driver dvipdfm . . . 10
3.5.3 Driver dvips . . . 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.

mehod=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 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 `\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

⟨*example1⟩

```
%<<END
\documentclass{article}
\usepackage{accsupp}[2007/11/14]
\usepackage{centernot}
% U+2226 NOT PARALLEL
% \mathrel{...} prevents page break in between
\newcommand*{\notparallel}{%
  \ensuremath{%
    \mathrel{%
      \BeginAccSupp{method=hex,unicode,ActualText=2226}%
      \centernot{\parallel}%
      \EndAccSupp{}%
    }%
  }%
}
\begin{document}
\begin{equation}
A\notparallel B
\end{equation}
\end{document}
%END
```

⟨/example1⟩

2.2 Example with pdfstringdef

⟨*example2⟩

```
%<<END
\documentclass{article}
\usepackage[unicode]{hyperref}
\usepackage{accsupp}[2007/11/14]
\begin{document}
  \begin{equation}
    \BeginAccSupp{
      method=pdfstringdef,
      unicode,
```

```
      ActualText={%
        a\texttwosuperior +b\texttwosuperior
        =c\texttwosuperior
      }
    }
    a^2 + b^2 = c^2
    \EndAccSupp{}
  \end{equation}
\end{document}
%END
```

⟨/example2⟩

3 Implementation

3.1 Package

⟨*package⟩

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname ACCSUPP@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\ACCSUPP@AtEnd{%
    \ACCSUPP@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{10}{12}% ^^J
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
```

```
\TMP@EnsureCode{94}{7}% ^ (superscript)
\TMP@EnsureCode{96}{12}% `
\TMP@EnsureCode{254}{12}% ^^fe
\TMP@EnsureCode{255}{12}% ^^ff
\edef\ACCSUPP@AtEnd{\ACCSUPP@AtEnd\noexpand\endinput}
```

Package identification.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{accsupp}%
  [2010/01/16 v0.3 Accessibility support by marked content (HO)]
\RequirePackage{pdfescape}[2007/02/25]
\RequirePackage{ifpdf}
\RequirePackage{ifxetex}
\RequirePackage{kvoptions}
\SetupKeyvalOptions{%
  family=ACCSUPP,%
  prefix=ACCSUPP@%
}
```

3.2 Space setup

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

3.3 Driver detection and setup

Driver declarations.

```
\def\ACCSUPP@DefineDriverKey{%
  \@dblarg\ACCSUPP@@DefineDriverKey
}
\def\ACCSUPP@@DefineDriverKey[#1]#2{%
  \define@key{ACCSUPP}{#2}[]{%
    \def\ACCSUPP@driver{#1}%
  }%
  \g@addto@macro\ACCSUPP@DisableOptions{%
    \DisableKeyvalOption{ACCSUPP}{#2}%
  }%
}
\let\ACCSUPP@DisableOptions\@empty
\ACCSUPP@DefineDriverKey{pdftex}
\ACCSUPP@DefineDriverKey{dvips}
\ACCSUPP@DefineDriverKey[dvips]{dvipsone}
\ACCSUPP@DefineDriverKey{dvipdfm}
\ACCSUPP@DefineDriverKey[dvipdfm]{dvipdfmx}
\ACCSUPP@DefineDriverKey[dvipdfm]{xetex}
\let\ACCSUPP@driver\relax
\InputIfFileExists{accsupp.cfg}{}{}
\providecommand*{\ActualTextDriverDefault}{dvips}
\ifpdf
  \def\ACCSUPP@driver{pdftex}%
\else
  \ifxetex
    \def\ACCSUPP@driver{dvipdfm}%
  \else
    \ifx\ACCSUPP@driver\relax
      \let\ACCSUPP@driver\ActualTextDriverDefault
    \fi
  \fi
\fi
```

Process options.

```
\ProcessKeyvalOptions*
```

```
\ACCSUPP@DisableOptions
```

Driver validation and loading.

```
\def\ACCSUPP@temp{pdftex}%
\ifpdf
  \ifx\ACCSUPP@temp\ACCSUPP@driver
  \else
    \PackageWarningNoLine{accsupp}{%
      Wrong driver ‘\ACCSUPP@driver’, using ‘pdftex’ instead%
    }%
    \let\ACCSUPP@driver\ACCSUPP@temp
  \fi
\else
  \ifx\ACCSUPP@temp\ACCSUPP@driver
    \PackageError{accsupp}{%
      Wrong driver, pdfTeX is not running in PDF mode.\MessageBreak
      Package loading is aborted%
    }\@ehc
    \expandafter\expandafter\expandafter\ACCSUPP@AtEnd
  \fi
  \def\ACCSUPP@temp{dvipdfm}%
  \ifxetex
    \ifx\ACCSUPP@temp\ACCSUPP@driver
    \else
      \PackageWarningNoLine{accsupp}{%
        Wrong driver ‘\ACCSUPP@driver’,\MessageBreak
        using ‘dvipdfm’ for XeTeX instead%
      }%
      \let\ACCSUPP@driver\ACCSUPP@temp
    \fi
  \fi
\fi%
\ifx\ACCSUPP@driver\relax
  \PackageError{accsupp}{%
    Missing driver option.\MessageBreak
    Package loading is aborted%
  }\@ehc
  \expandafter\ACCSUPP@AtEnd
\fi%
\InputIfFileExists{accsupp-\ACCSUPP@driver.def}{}{%
  \PackageError{accsupp}{%
    Unsupported driver ‘\ACCSUPP@driver’.\MessageBreak
    Package loading is aborted%
  }\@ehc
  \ACCSUPP@AtEnd
}%
```

3.4 Main macro

```
\DeclareBoolOption{unicode}
\DeclareStringOption[page]{pdfliteral}
\DeclareStringOption{Lang}
\def\ACCSUPP@method{escape}
\define@key{ACCSUPP}{method}{%
  \@ifundefined{ACCSUPP@method@#1}{%
    \PackageError{accsupp}{%
      Ignoring unknown method ‘#1’%
    }\@ehc
  }{%
    \edef\ACCSUPP@method{#1}%
  }%
}
```

```
\let\ACCSUPP@Lang\relax
\def\ACCSUPP@temp#1{%
  \expandafter\ACCSUPP@@temp\csname ACCSUPP@#1\endcsname{#1}%
}
\def\ACCSUPP@@temp#1#2{%
  \let#1\relax
  \define@key{ACCSUPP}{#2}{%
    \def#1{##1}%
    \ifx#1\@empty
      \def#1{()}%
    \else
      \csname ACCSUPP@method@\ACCSUPP@method\endcsname#1%
    \fi
  }%
}
\ACCSUPP@temp{Alt}
\ACCSUPP@temp{ActualText}
\ACCSUPP@temp{E}
\newcommand*{\BeginAccSupp}[1]{%
  \begingroup
    \setkeys{ACCSUPP}{#1}%
    \edef\ACCSUPP@span{%
      /Span<<%
        \ifx\ACCSUPP@Lang\relax
        \else
          /Lang\ACCSUPP@Lang
        \fi
        \ifx\ACCSUPP@Alt\relax
        \else
          /Alt\ACCSUPP@Alt
        \fi
        \ifx\ACCSUPP@ActualText\relax
        \else
          /ActualText\ACCSUPP@ActualText
        \fi
        \ifx\ACCSUPP@E\relax
        \else
          /E\ACCSUPP@E
        \fi
      >>%
    }%
    \ACCSUPP@bdc
    \ACCSUPP@space
  \endgroup
}
\newcommand*{\EndAccSupp}[1]{%
  \begingroup
    \setkeys{ACCSUPP}{#1}%
    \ACCSUPP@emc
  \endgroup
}
\def\ACCSUPP@space{%
  \ifACCSUPP@space
    \begingroup
      \@ifundefined{ACCSUPP@Font}{%
        \global\font\ACCSUPP@Font=\ACCSUPP@spacefont\relax
      }{}%
      \leavevmode
      \setbox\z@\hbox{\ACCSUPP@Font\char\ACCSUPP@spacechar}%
      \wd\z@\z@
      \ht\z@\z@
      \dp\z@\z@
```

```
      \copy\z@
    \endgroup
  \fi
}
\newcommand*{\AccSuppSetup}{%
  \setkeys{ACCSUPP}%
}
```

3.4.1 Input methods

```
\def\ACCSUPP@method@plain#1{%
  \csname @safe@activestrue\endcsname
  \edef#1{%
    (%
    \ifACCSUPP@unicode
      \string\376\string\377%
    \fi
    #1%
    )%
  }%
  \@onelevel@sanitize#1%
}
\def\ACCSUPP@method@escape#1{%
  \EdefEscapeString#1{%
    \ifACCSUPP@unicode
      ^^fe^^ff%
    \fi
    #1%
  }%
  \edef#1{(#1)}%
}%
\def\ACCSUPP@method@hex#1{%
  \edef#1{%
    <%
    \ifACCSUPP@unicode
      FEFF%
    \fi
    #1%
    >%
  }%
}
\def\ACCSUPP@method@pdfstringdef#1{%
  \ifACCSUPP@unicode
    \@ifundefined{hypersetup}{}{%
      \hypersetup{unicode}%
    }%
  \fi
  \@ifundefined{pdfstringdef}{%
    \PackageError{accsupp}{%
      Method `pdfstringdef' requires package `hyperref'%
    }\@ehc
    \let\ACCSUPP@temp\@empty
  }{%
    \begingroup
      \setbox0=\hbox{%
        \pdfstringdef\ACCSUPP@temp#1%
        \global\let\ACCSUPP@temp\ACCSUPP@temp
      }%
    \endgroup
  }%
  \edef#1{(\ACCSUPP@temp)}%
}
```

```
\ACCSUPP@AtEnd%
⟨/package⟩
```

3.5 Drivers

3.5.1 Driver pdftex

```
⟨*pdftex⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesFile{accsupp-pdftex.def}%
  [2010/01/16 v0.3 accsupp driver for pdfTeX (HO)]%
\def\ACCSUPP@bdc{%
  \pdfliteral\ACCSUPP@pdfliteral{\ACCSUPP@span BDC}%
}
\def\ACCSUPP@emc{%
  \pdfliteral\ACCSUPP@pdfliteral{EMC}%
}
⟨/pdftex⟩
```

3.5.2 Driver dvipdfm

```
⟨*dvipdfm⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesFile{accsupp-dvipdfm.def}%
  [2010/01/16 v0.3 accsupp driver for dvipdfm (HO)]%
\def\ACCSUPP@bdc{%
  \special{pdf:content \ACCSUPP@span BDC}%
}
\def\ACCSUPP@emc{%
  \special{pdf:content EMC}%
}
⟨/dvipdfm⟩
```

3.5.3 Driver dvips

```
⟨*dvips⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesFile{accsupp-dvips.def}%
  [2010/01/16 v0.3 accsupp driver for dvips (HO)]%
\def\ACCSUPP@bdc{%
  \special{ps:[\ACCSUPP@span/BDC pdfmark}%
}
\def\ACCSUPP@emc{%
  \special{ps:[/EMC pdfmark}%
}
⟨/dvips⟩
```

4 Test

4.1 Catcode checks for loading

```
⟨*test1⟩
\NeedsTeXFormat{LaTeX2e}
\documentclass{minimal}
\makeatletter
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
```

```
  \advance\count@\@ne
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@\@ne
  \repeat
}
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{127}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
  \catcode13=5 %
  \RequirePackage{accsupp}[2010/01/16]\relax
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/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
```

[1] ftp://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):

```
accsupp.sty             → tex/latex/oberdiek/accsupp.sty
accsupp-pdftex.def      → tex/latex/oberdiek/accsupp-pdftex.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

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 pdfLaTeX:

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

Symbols

\# 388
\% 391
\@ 384
\@dblarg 111
\@ehc 158, 177, 184, 195, 309
\@empty 121, 208, 310
\@ifundefined 192, 254, 302, 306
\@ne 368, 376
\@onelevel@sanitize 279
\[. 389
\\ 385
\{ 386
\} 387
\] 390

Numbers

\3 274
\␣ 392

A

\ACCSUPP@@DefineDriverKey . . 111, 113
\ACCSUPP@@temp 202, 204
\ACCSUPP@ActualText 231, 233
\ACCSUPP@Alt 227, 229
\ACCSUPP@AtEnd 72, 73, 95, 159, 178, 185, 321
\ACCSUPP@bdc 241, 327, 338, 349
\ACCSUPP@DefineDriverKey 110, 122, 123, 124, 125, 126, 127

\ACCSUPP@DisableOptions 117, 121, 143
\ACCSUPP@driver ... 115, 128, 132, 135, 137, 138, 146, 149, 151, 154, 163, 166, 169, 173, 180, 182
\ACCSUPP@E ... 235, 237
\ACCSUPP@emc ... 248, 330, 341, 352
\ACCSUPP@Font ... 255, 258
\ACCSUPP@Lang ... 200, 223, 225
\ACCSUPP@method ... 190, 197, 211
\ACCSUPP@method@escape ... 281
\ACCSUPP@method@hex ... 290
\ACCSUPP@method@pdfstringdef ... 300
\ACCSUPP@method@plain ... 269
\ACCSUPP@pdfliteral ... 328, 331
\ACCSUPP@space ... 242, 251
\ACCSUPP@spacechar ... 258
\ACCSUPP@spacefont ... 255
\ACCSUPP@span ... 221, 328, 339, 350
\ACCSUPP@temp ... 144, 146, 151, 154, 161, 163, 169, 201, 215, 216, 217, 310, 314, 315, 319
\AccSuppSetup ... 266
\ActualTextDriverDefault ... 130, 138
\advance ... 368, 376

B

\begin ... 17, 18, 29, 30
\BeginAccSupp ... 2, 11, 31, 218

C

\catcode ... 46, 47, 49, 50, 51, 55, 56, 57, 58, 59, 60, 61, 64, 65, 67, 68, 69, 70, 74, 76, 365, 374, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393
\centernot ... 12
\char ... 258
\copy ... 262
\count@ ... 361, 365, 367, 368, 372, 374, 375, 376
\csname ... 53, 202, 211, 270, 398

D

\DeclareBoolOption ... 107, 187
\DeclareStringOption 108, 109, 188, 189
\define@key ... 114, 191, 206
\DisableKeyvalOption ... 118
\documentclass ... 3, 26, 358
\dp ... 261

E

\EdefEscapeString ... 282
\end ... 20, 21, 41, 42, 399
\EndAccSupp ... 2, 13, 40, 245
\endcsname ... 53, 202, 211, 270, 398
\endinput ... 95
\endlinechar ... 48, 54, 66
\ensuremath ... 9

F

\font ... 255

G

\g@addto@macro ... 117

H

\hbox ... 258, 313
\ht ... 260
\hypersetup ... 303

I

\ifACCSUPP@space ... 252
\ifACCSUPP@unicode ... 273, 283, 293, 301
\ifnum ... 367, 375
\ifpdf ... 131, 145
\ifx ... 137, 146, 154, 163, 173, 208, 223, 227, 231, 235
\ifxetex ... 134, 162
\InputIfFileExists ... 129, 180

L

\leavevmode ... 257
\loop ... 362, 373

M

\makeatletter ... 359
\mathrel ... 7, 10
\MessageBreak ... 156, 166, 175, 182

N

\NeedsTeXFormat 96, 324, 335, 346, 357
\newcommand ... 8, 218, 245, 266
\notparallel ... 8, 19

P

\PackageError ... 155, 174, 181, 193, 307
\PackageWarningNoLine ... 148, 165
\parallel ... 12
\pdfliteral ... 328, 331
\pdfstringdef ... 314
\ProcessKeyvalOptions ... 142
\providecommand ... 130
\ProvidesFile ... 325, 336, 347
\ProvidesPackage ... 97

R

\RangeCatcodeInvalid ... 371, 380, 381, 382, 383
\repeat ... 369, 377
\RequirePackage 99, 100, 101, 102, 394
\RestoreCatcodes ... 360, 363, 364, 395

S

\setbox ... 258, 313
\setkeys ... 220, 247, 267
\SetupKeyvalOptions ... 103
\special ... 339, 342, 350, 353

T

\Test ... 379, 397
\texttwosuperior ... 35, 36
\the 54, 55, 56, 57, 58, 59, 60, 61, 74, 365
\TMP@EnsureCode ... 71, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94

U

\usepackage ... 4, 5, 27, 28

W

\wd 259

X

\x 52, 64

Z

\z@ 258, 259, 260, 261, 262