

README for hyperref bundle

Heiko Oberdiek, Sebastian Rahtz

2012/07/12

Contents

1	Introduction	3
2	Download	3
	Sources	3
3	Installation	4
	Installation with ZIP file in TDS format	4
	Manual installation	4
4	Additional Packages	4
5	New Features	6
	Option ‘psdextra’	6
	\IfHyperBooleanExists and \IfHyperBoolean	6
	\unichar	6
	\ifpdfstringunicode	6
	Customizing index style file with \nohyperpage	7
	Expert feature \HyperDestNameFilter	8
	Experimental option ‘ocgcolorlinks’	8
	Option ‘pdfa’	8
	Option ‘linktoc’ added	9
	Option ‘pdfnewwindow’ changed	9
	Flag options for PDF forms	10
	Option ‘pdfversion’	11
	Field option ‘name’	11
	Option ‘pdfencoding’	11
	Color options/package hycolor	12
	Option pdfusetitle	12
	Starred form of \autoref	12
	Link border style	12
	Option ”bookmarksdepth”	13
	Option ”pdfescapeform”	13
	Default driver setting	13
	Backref entries	14
	\phantomsection	15
	\hypercalcbp	16
6	Package Compatibility	16
	algorithm	16
	amsmath	16
	amsrefs	16
	arydshln, longtable	16
	babel/magyar.ldf	17

babel/spanish.ldf	17
bibentry	17
bigfoot	17
chappg	17
cite	17
count1to	18
dblaccnt	18
easyeqn	18
ellipsis	18
float	18
endnotes	18
foiltex	18
footnote	19
geometry	19
IEEEtran.cls	19
index	19
lastpage	19
linguex	20
ltabptch	20
mathenv	20
minitoc-hyper	20
multind	20
natbib	20
nomencl	20
prettyref	20
ntheorem	21
setspace	21
sidecap	21
subfigure	21
titleref	21
tabularx	22
titlesec	22
ucs/utf8x.def	22
varioref	22
verse	22
vietnam	23
XeTeX	23
7 Limitations	24
Wrapped/broken link support	24
Links across pages	25
Footnotes	25
8 Hints	25
Spaces in option values	25
Index with makeindex	25
Warning "bookmark level for unknown <foobar> defaults to 0"	26
Link anchors in figures	26
Additional unicode characters in bookmarks and pdf information entries:	26
Footnotes	27
Subordinate counters	28
9 Authors/Maintainers	29

10 Bug Reports	29
Bug address	29
Vietnamese part	29
Arabic part	29
11 Known Problems	30
12 ToDo	30
13 Versions in TeX distributions	30
TeX Live	30

1 Introduction

This package is used to emend cross-referencing commands in LaTeX to produce some sort of \special commands; there are backends for the \special set defined for HyperTeX dvi processors, for embedded pdfmark commands for processing by Acrobat Distiller (dvips and dvipsone), for dviwindo, for pdfTeX, for dvipdfm, for TeX4ht, and for VTeX's pdf and HTML backends.

Included are:

- a) 'backref' a package by David Carlisle to provide links back from bibliography to the main text; these are hypertext links after using hyperref.
- b) nameref' a package to allow reference to the *names* of sections rather than their numbers.

2 Download

'hyperref' is available on CTAN:

CTAN:macros/latex/contrib/hyperref/

Also a ZIP file is provided that contains the files, already sorted in a TDS tree:

CTAN:install/macros/latex/contrib/hyperref.tds.zip

'CTAN:' means one of the 'Comprehensive TeX Archive Network' nodes or one of its mirrors. This is explained in

<http://www.tex.ac.uk/cgi-bin/texfaq2html?label=archives>

The main repository of hyperref is located at the server of the international 'TeX User Group' (TUG):

<http://www.tug.org/applications/hyperref/>

The CTAN stuff will be mirrored automatically from the ftp server, so

<ftp://ftp.tug.org/pub/tex/hyperref/>

corresponds to

CTAN:macros/latex/contrib/hyperref/

Sources

The subversion source repository for hyperref is hosted at Puszcz:

<https://puszcza.gnu.org.ua/svn/?group=hyperref>

It offers anonymous access. For daily use I recommend using

a release. If new features are added or bugs are fixed, then the new version is usually released fast. Please avoid distributing intermediate states between releases.

3 Installation

Installation with ZIP file in TDS format

The ZIP file 'hyperref.tds.zip' contains the files sorted in a TDS tree. Thus you can directly unpack the ZIP file inside a TDS tree. (See CTAN:tds.zip for an explanation of TDS.)

Example:

```
cd /...somewhere.../texmf
unzip /...downloadpath.../hyperref.tds.zip
```

Do not forget to refresh the file name database of this TDS tree,

Example:

```
texhash /...somewhere.../texmf
```

Manual installation

- a) Download the hyperref files from CTAN or the TUG server.
If necessary, unpack them.
- b) If directory 'beta' exists, replace the files by the counterparts in this directory, if you want to use the latest versions.
- c) Generate the package and driver files:
tex hyperref.ins
- d) Install the files '*.sty', '*.def', and '*.cfg' in your TDS tree:
cp *.sty *.def *.cfg TDS:tex/latex/hyperref/
Replace 'TDS:' by the prefix of your TDS tree (texmf directory).
The exception is bmhydoc.sty, it belongs to the source files (TDS:source/latex/hyperref/).
- e) Copy the documentation files to "TDS:doc/latex/hyperref/":
manual.pdf, README, README.pdf, ChangeLog, ChangeLog.pdf, slides.pdf, paper.pdf, options.pdf, hyperref.pdf, backref.pdf, nameref.pdf (Also the HTML version of the manual can be put there.)
- f) Update the databases if necessary, eg. for teTeX:
mktexlsr .../texmf

4 Additional Packages

Depending on the driver and option settings, hyperref loads other packages:

- * atbegshi.sty:
CTAN:macros/latex/contrib/oberdiek/atbegshi.pdf
- * atveryend.sty:
CTAN:macros/latex/contrib/oberdiek/atveryend.pdf
- * backref.sty (loaded by option 'backref' or 'pagebackref'):
CTAN:macros/latex/contrib/hyperref/backref.dtx
- * bigintcalc.sty (loaded by package 'bitset'):
CTAN:macros/latex/contrib/oberdiek/bigintcalc.pdf
- * bitset.sty:
CTAN:macros/latex/contrib/oberdiek/bitset.pdf
- * color.sty (loaded by option 'colorlinks'):
CTAN:macros/latex/required/graphics/

- * `etexcmds.sty`:
`CTAN:macros/latex/contrib/oberdiek/etexcmds.pdf`
- * `getttitlestring.sty` (loaded by package ‘`nameref`’):
`CTAN:macros/latex/contrib/oberdiek/getttitlestring.pdf`
- * `hycolor.sty`:
`CTAN:macros/latex/contrib/oberdiek/hycolor.pdf`
- * `infwarerr.sty` (loaded by packages ‘`etexcmds`’, ‘`stringenc`’, ‘`atbegshi`’, ‘`bitset`’):
`CTAN:macros/latex/contrib/oberdiek/infwarerr.pdf`
- * `intcalc.sty`:
`CTAN:macros/latex/contrib/oberdiek/intcalc.pdf`
- * `ifpdf.sty` (loaded by package ‘`atbegshi`’):
`CTAN:macros/latex/contrib/oberdiek/atbegshi.pdf`
- * `ifluatex.sty` (loaded by package ‘`pdftexcmds`’):
`CTAN:macros/latex/contrib/oberdiek/ifluatex.pdf`
- * `ifvtex.sty`:
`CTAN:macros/latex/contrib/oberdiek/ifvtex.pdf`
- * `ifxetex.sty`:
`CTAN:macros/generic/ifxetex/ifxetex.sty`
- * `intcalc.sty` (loaded by package ‘`bitset`’):
`CTAN:macros/latex/contrib/oberdiek/intcalc.pdf`
- * `keyval.sty`
`CTAN:macros/latex/required/graphics/`
- * `kvoptions.sty`:
`CTAN:macros/latex/contrib/oberdiek/kvoptions.pdf`
- * `kvsetkeys.sty`:
`CTAN:macros/latex/contrib/oberdiek/kvsetkeys.pdf`
- * `letltxmacro.sty`:
`CTAN:macros/latex/contrib/oberdiek/letltxmacro.pdf`
- * `ltxcmds.sty` (loaded by package ‘`pdftexcmds`’):
`CTAN:macros/latex/contrib/oberdiek/ltxcmds.pdf`
- * `memhfixc.sty` (loaded if class ‘`memoir`’ is loaded):
`CTAN:macros/latex/contrib/memoir/memhfixc.sty`
- * `nameref.sty`:
`CTAN:macros/latex/contrib/hyperref/nameref.dtx`
- * `pdfescape.sty` (loaded by package ‘`stringenc`’):
`CTAN:macros/latex/contrib/oberdiek/pdfescape.pdf`
- * `pdftexcmds.sty`
`CTAN:macros/latex/contrib/oberdiek/pdftexcmds.pdf`
- * `refcount.sty` (loaded by package ‘`nameref`’)
`CTAN:macros/latex/contrib/oberdiek/refcount.pdf`
- * `rerunfilecheck.sty`:
`CTAN:macros/latex/contrib/oberdiek/rerunfilecheck.pdf`
- * `stringenc.sty`:
`CTAN:macros/latex/contrib/oberdiek/stringenc.pdf`
- * `tex4ht.sty` (loaded by option ‘`tex4ht`’):
`CTAN:support/TeX4ht/`
- * `uniquecounter.sty` (loaded by package ‘`rerunfilecheck`’):
`CTAN:macros/latex/contrib/oberdiek/uniquecounter.pdf`
- * `url.sty`
`CTAN:macros/latex/contrib/other/misc/url.sty`
- * `vtexhtml.sty` (loaded if VTeX is used in HTML mode)
- * `xcolor-patch.sty` (loaded by package ‘`hycolor`’)
`CTAN:macros/latex/contrib/oberdiek/hycolor.pdf`

5 New Features

Option ‘psdextra’

LaTeX’s NFSS is used to assist the conversion of arbitrary TeX strings to PDF strings (bookmarks, PDF information entries). Many math command names (`\geq`, `\notin`, ...) are not in control of NFSS, therefore they are defined with prefix ‘text’ (`\textgeq`, `\textnotin`, ...). They can be mapped to short names during the processing to PDF strings. The disadvantage is that they are many hundreds macros that need to be redefined for each PDF string conversion. Therefore this can be enabled or disabled as option ‘psdextra’. On default the option is turned off (set to ‘false’). Turning the option on means that the short names are available. Then `\geq` can directly be used instead of `\textgeq`.

`\IfHyperBooleanExists` and `\IfHyperBoolean`

`\IfHyperBooleanExists{OPTION}{YES}{NO}`

If a hyperref OPTION is a boolean, that means it takes values ‘true’ or ‘false’, then `\IfHyperBooleanExists` calls YES, otherwise NO.

`\IfHyperBoolean{OPTION}{YES}{NO}`

Macro `\IfHyperBoolean` calls YES, if OPTION exists as boolean and is enabled. Otherwise NO is executed.

Both macros are expandable. Additionally option ‘stopppedearly’ is available. It is enabled if `\MaybeStopEarly` or `\MaybeStopNow` end hyperref prematurely.

`\unichar`

If a Unicode character is not supported by `puenc.def`, it can be given by using `\unichar`. It’s name and syntax is inherited from package ‘ucs’. However it is defined independently for use in hyperref’s `\pdfstringdef` (that converts arbitrary TeX code to PDF strings or tries to do this).

Macro `\unichar` takes a TeX number as argument, examples for U+263A (WHITE SMILING FACE):

`\unichar{"263A}% hexadecimal notation`

`\unichar{9786}% decimal notation`

“” must not be a babel shorthand character or otherwise active. Otherwise prefix it with `\string`:

`\unichar{\string"263A}% converts “” to “” with catcode 12 (other)`

Users of (n)german packages or babel options may use `\dq` instead:

`\unichar{\dq 263A}% \dq is double quote with catcode 12 (other)`

`\ifpdfstringunicode`

Some features of the PDF specification needs PDF strings. Examples are bookmarks or the entries in the information dictionary. The PDF specification allows two encodings ‘PDFDocEncoding’ (8-bit encoding) and ‘Unicode’ (UTF-16). The user can help using `\texorpdfstring` to replace complicate TeX constructs by a representation for the PDF string. However `\texorpdfstring`

does not distinguish the two encodings. This gap closes `\ifpdfstringunicode`. It is only allowed in the second argument of `\texorpdfstring` and takes two arguments, the first allows the full range of Unicode. The second is limited to the characters available in `PDFDocEncoding`.

As example we take a macro definition for the Vietnamese name of Han The Thanh. Correctly written it needs some accented characters, one character even with a double accent. Class ‘`tugboat.cls`’ defines a macro for the typesetted name:

```
\def\Thanh{%
  H\`an~%
  Th\`e\llap{\raise 0.5ex\hbox{\`{}}}%
  ~Th\`anh%
}
```

It’s not entirely correct, the second accent over the ‘e’ is not an acute, but a hook. However standard LaTeX does not provide such an accent.

Now we can extend the definition to support `hyperref`. The first and the last word are already supported automatically. Characters with two or more accents are a difficult business in LaTeX, because the NFSS2 macros of the LaTeX kernel do not support more than one accent. Therefore also `puenc.def` misses support for them. But we can provide it using `\unichar`. The character in question is:

```
% U+1EC3 LATIN SMALL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
```

Thus we can put this together:

```
\def\Thanh{%
  H\`an~%
  \texorpdfstring{Th\`e\llap{\raise 0.5ex\hbox{\`{}}}}%
  {\ifpdfstringunicode{Th\unichar{"1EC3}}{Th\`e}}%
  ~Th\`anh%
}
```

For `PDFDocEncoding` (PD1) the variant above has dropped the second accent. Alternatively we could provide a representation without accents instead of wrong accents:

```
\def\Thanh{%
  \texorpdfstring{%
    H\`an~%
    Th\`e\llap{\raise 0.5ex\hbox{\`{}}}}%
    ~Th\`anh%
  }{%
    \ifpdfstringunicode{%
      H\`an Th\unichar{"1EC3} Th\`anh%
    }{%
      Han The Thanh%
    }%
  }%
}
```

Customizing index style file with `\nohyperpage`

Since version 2008/08/14 v6.78f.

For hyperlink support in the index, `hyperref` inserts `\hyperpage` into the index macros. After processing with `Makeindex`, `\hyperpage`

analyzes its argument to detect page ranges and page comma lists. However, only the standard settings are supported directly:

```
delim_r "--"
delim_n ", "
```

(See manual page/documentation of Makeindex that explains the keys that can be used in style files for Makeindex.)

Customized versions of

```
delim_r, delim_n, suffix_2p, suffix_3p, suffix_mp
```

needs markup that `\hyperpage` can detect and knows that this stuff does not belong to a page number. Makro `\nohyperpage` serves as this markup. Put the customized code for these keys inside `\nohyperpage`, e.g.:

```
suffix_2p "\nohyperpage{f.}"
suffix_3p "\nohyperpage{ff.}"
```

(Depending on the typesetting tradition some space `"\,"` or `"~"` should be put before the first `f` inside `\nohyperpage`.)

Expert feature `\HyperDestNameFilter`

Since version 2008/07/16 v6.78c.

Each anchor name and inner link name is put through filter `\HyperDestNameFilter`. The default behaviour is to let the name unchanged:

```
\def\HyperDestNameFilter#1{#1}
```

It can be redefined to prepend or append something, to avoid name collisions if several files are merged together.

For example, two documents with `hyperref` can be generated using different prefixes for destination names. Then it is easier to merge them with `'dviconcat'` or other merge utilities.

Experimental option `'ocgcolorlinks'`

The idea are colored links, when viewed, but printed without colors. This new experimental option `'ocgcolorlinks'` uses Optional Content Groups, a feature introduced in PDF 1.5.

* The option must be given for package loading:

```
\usepackage[ocgcolorlinks]{hyperref}
```

* Main disadvantage: Links cannot be broken across lines.

PDF reference 1.7: 4.10.2 "Making Graphical Content Optional":
Graphics state operations, such as setting the color, ..., are still applied.

Therefore the link text is put in a box and set twice, with and without color.

* The feature can be switched off by `\hypersetup{ocgcolorlinks=false}` inside the document.

* Supported drivers: `pdftex`, `dvipdfm`

* The PDF version should be at least 1.5. It is automatically set for `pdfTeX`. Users of `dvipdfmx` set the version on the command line: `dvipdfmx -V 5`

Option `'pdfa'`

The new option `'pdfa'` tries to avoid violations of PDF/A in code

generated by hyperref. However, the result is usually not in PDF/A, because many features aren't controlled by hyperref (XMP metadata, fonts, colors, driver dependend low level stuff, ...).

Currently, option 'pdfa' sets and disables the following items:

- * Enabled annotation flags: Print, NoZoom, NoRotate [PDF/A 6.5.3].
- * Disabled annotation flags: Hidden, Invisible, NoView [PDF/A 6.5.3].
- * Disabled: Launch action (`\href{run:...}`) [PDF/A 6.6.1].
- * Restricted: Named actions (`\Acrobatmenu: NextPage, PrevPage, FirstPage, LastPage`) [PDF/A 6.6.1].
- * Many things are disabled in PDF formulators:
 - * JavaScript actions [PDF/A 6.6.1]
 - * Trigger events (additional actions) [PDF/A 6.6.2]
 - * Push button (because of JavaScript)
 - * Interactive Forms: Flag `NeedAppearances` is the default 'false' (Because of this, hyperref's implementation of Forms looks ugly). [PDF/A 6.9]

The default value of the new option 'pdfa' is 'false'. It influences the loading of the package and cannot be changed after hyperref is loaded (`\usepackage{hyperref}`).

ToDo:

- * XMP support
- * ...

But perhaps Adobe Acrobat is now happy and can now convert the PDF file to PDF/A.

Option 'linktoc' added

The new option 'linktoc' allows more control which part of an entry in the table of contents is made into a link:

- * 'linktoc=none' (no links)
- * 'linktoc=section' (default behaviour, same as 'linktocpage=false')
- * 'linktoc=page' (same as 'linktocpage=true')
- * 'linktoc=all' (both the section and page part are links)

Option 'pdfnewwindow' changed

Before 6.77b:

- * `pdfnewwindow=true` --> `/NewWindow true`
- * `pdfnewwindow=false` --> (absent)
- * unused `pdfnewwindow` --> (absent)

Since 6.77b:

- * `pdfnewwindow=true` --> `/NewWindow true`
- * `pdfnewwindow=false` --> `/NewWindow false`
- * `pdfnewwindow={}` --> (absent)
- * unused `pdfnewwindow` --> (absent)

Rationale: There is a difference between setting to 'false' and an absent entry. In the former case the new document replaces the old one, in the latter case the PDF viewer application should respect the user preference.

Flag options for PDF forms

PDF form field macros (\TextField, \CheckBox, ...) support boolean flag options. The option name is the lowercase version of the names in the PDF specification (1.7):

http://www.adobe.com/devnet/pdf/pdf_reference.html

http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference.pdf

Options (convert to lowercase) except flags in square brackets:

* Table 8.16 Annotation flags (page 608):

- 1 Invisible
- 2 Hidden (PDF 1.2)
- 3 Print (PDF 1.2)
- 4 NoZoom (PDF 1.3)
- 5 NoRotate (PDF 1.3)
- 6 NoView (PDF 1.3)
- [7 ReadOnly (PDF 1.3)] ignored for widget annotations, see table 8.70
- 8 Locked (PDF 1.4)
- 9 ToggleNoView (PDF 1.5)
- 10 LockedContents (PDF 1.7)

* Table 8.70 Field flags common to all field types (page 676):

- 1 ReadOnly
- 2 Required
- 3 NoExport

* Table 8.75 Field flags specific to button fields (page 686):

- 15 NoToggleToOff (Radio buttons only)
- 16 Radio (set: radio buttons, clear: check box, pushbutton: clear)
- 17 Pushbutton
- 26 RadiosInUniso (PDF 1.5)

* Table 8.77 Field flags specific to text fields (page 691):

- 13 Multiline
- 14 Password
- 21 FileSelect (PDF 1.4)
- 23 DoNotSpellCheck (PDF 1.4)
- 24 DoNotScroll (PDF 1.4)
- 25 Comb (PDF 1.5)
- 26 RichText (PDF 1.5)

* Table 8.79 Field flags specific to choice fields (page 693):

- 18 Combo (set: combo box, clear: list box)
- 19 Edit (only useful if Combo is set)
- 20 (Sort) for authoring tools, not PDF viewers
- 22 MultiSelect (PDF 1.4)
- 23 DoNotSpellCheck (PDF 1.4) (only useful if Combo and Edit are set)
- 27 CommitOnSelChange (PDF 1.5)

* Table 8.86 Flags for submit-form actions (page 704):

- [1 Include/Exclude] unsupported, use 'noexport' (table 8.70) instead
- 2 IncludeNoValueFields
- [3 ExportFormat] handled by option 'export'
- 4 GetMethod
- 5 SubmitCoordinates
- [6 XFDF (PDF 1.4)] handled by option 'export'
- 7 IncludeAppendSaves (PDF 1.4)
- 8 IncludeAnnotations (PDF 1.4)
- [9 SubmitPDF (PDF 1.4)] handled by option 'export'
- 10 CanonicalFormat (PDF 1.4)
- 11 ExclNonUserAnnots (PDF 1.4)

- 12 ExclFKey (PDF 1.4)
- 14 EmbedForm (PDF 1.5)

New option ‘export’ sets the export format of a submit action.

Valid values are (upper- or lowercase):

- * FDF
- * HTML
- * XFDF
- * PDF (not supported by Acrobat Reader)

Option ‘pdfversion’

This is an experimental option. It notifies ‘hyperref’ about the intended PDF version. Currently this is used in code for PDF forms (implementation notes 116 and 122 of PDF spec 1.7).

Values: 1.2, 1.3, 1.4, 1.5, 1.6, 1.7. Values below 1.2 are not supported, because most drivers expect higher PDF versions.

The option must be used early, not after `\usepackage{hyperref}`.

In theory this option should also set the PDF version, but this is not generally supported.

- * pdfTeX below 1.10a: unsupported.
- pdfTeX \geq 1.10a and $<$ 1.30: `\pdfoptionpdfminorversion`
- pdfTeX \geq 1.30: `\pdfminorversion`
- * dvipdfm: configuration file, example:
TeX Live 2007, `texmf/dvipdfm/config/config`, entry ‘V 2’.
- * dvipdfmx: configuration file, example:
TeX Live 2007, `texmf/dvipdfm/dvipdfmx.cfg`, entry ‘V 4’.
- * Ghostscript: option `-dCompatibilityLevel` (this is set in ‘ps2pdf12’, ‘ps2pdf13’, ‘ps2pdf14’).

The current PDF version is used as default if this version can be detected (only pdfTeX \geq 1.10a). Otherwise the lowest version 1.2 is assumed. Thus ‘hyperref’ tries to avoid PDF code that breaks this version, but is free to use ignorable higher PDF features.

Field option ‘name’

Many form objects uses the label argument for several purposes:

- * Laid out label.
- * As name in HTML structures.

Code that is suitable for layouting with TeX can break in the structures of the output format.

If option ‘name’ is given, then its value is used as name in the different output structures. Thus the value should consist of letters only.

Option ‘pdfencoding’

The PDF format allows two encodings for bookmarks and entries in the information dictionary: PDFDocEncoding and Unicode as UTF-16BE. Option “pdfencoding” selects between these encodings:

- * “pdfdoc” uses PDFDocEncoding. It uses just one byte per character,

but the supported characters are limited (244 in PDF-1.7).

- * "unicode" sets Unicode. It is encoded as UTF-16BE. Two bytes are used for most characters, surrogates need four bytes.
- * "auto" PDFDocEncoding if the string does not contain characters outside the encoding and Unicode otherwise.

Color options/package hycolor

See documentation of package 'hycolor'.

Option pdfusetitle

If option pdfusetitle is set then hyperref tries to derive the values for pdftitle and pdfauthor from \title and \author. An optional argument for \title and \author is supported (class amsart).

Starred form of \autoref

\autoref* generates a reference without link as \ref* or \pageref*.

Link border style

Links can be underlined instead of the default rectangle or options "colorlinks", "frenchlinks". This is done by option pdfborderstyle={/S/U/W 1}

Some remarks:

- * AR7/Linux seems to have a bug, that don't use the default value "1" for the width, but zero, thus that the underline is not visible without "/W 1". The same applies for dashed boxes, eg.:

```
pdfborderstyle={/S/D/D[3 2]/W 1}
```

- * The syntax is described in the PDF specification, look for "border style", eg. Table 8.13 "Entries in a border style dictionary" (specification for version 1.6)

- * The border style is removed by pdfborderstyle={} This is automatically done if option colorlinks is enabled.

- * Be aware that not all PDF viewers support this feature, not even Acrobat Reader itself:

Some support:

- * AR7/Linux: "underline" and "dashed", but the border width must be given.
- * xpdf 3.00: "underline" and "dashed"

Unsupported:

- * AR5/Linux
- * ghostscript 8.50

Option "bookmarksdepth"

The depth of the bookmarks can be controlled by the new option "bookmarksdepth". The option acts globally and distinguishes three cases:

- * "bookmarksdepth" without value
Then hyperref uses the current value of counter "tocdepth".
This is the compatible behaviour and the default.
- * "bookmarksdepth=<number>", the value is number (also negative):
The depth for the bookmarks are set to this number.
- * "bookmarksdepth=<name>"
The <name> is a document division name (part, chapter, ...).
It must not start with a digit or minus to avoid mixing up with the number case. Internally hyperref uses the value of macro "\toclevel@<name>".

Examples:

```
\hypersetup{bookmarksdepth=paragraph}
\hypersetup{bookmarksdepth=4} % same as before
\hypersetup{bookmarksdepth} % counter "tocdepth" is used
```

Option "pdfescapeform"

There are many places where arbitrary strings end up as PS or PDF strings. The PS/PDF strings in parentheses form require the protection of some characters, e.g. unmatched left or right parentheses need escaping or the escape character itself (backslash).

Since 2006/02/12 v6.75a the PS/PDF driver should do this automatically. However I assume a problem with compatibility, especially regarding the form part where larger amounts of JavaScript code can be present. It would be a pain to remove all the escaping, because an additional escaping layer can falsify the code.

Therefore a new option pdfescapeform was introduced:

- * pdfescapeform=false
Escaping for the formulars are disabled, this is the compatibility behaviour, therefor this is the default.
- * pdfescapeform=true
Then the PS/PDF drivers do all the necessary escaping. This is the logical choice and the recommended setting. For example, the user writes JavaScript as JavaScript and do not care about escaping characters for PS/PDF output.

Default driver setting

(hyperref >= 6.72s)

If no driver is given, hyperref tries its best to guess the most suitable driver. Thus it loads "hpdfTeX", if pdfTeX is detected running in PDF mode. Or it loads the corresponding VTeX driver for VTeX's working modes.

Unhappily many driver programs run after the TeX compiler, so hyperref does not have a chance (dvips, dvipdfm, ...). In this case driver "hypertex" is loaded that supports the HyperTeX features that are recognized by xdvi for example.

This behaviour, however, can easily be changed in the configuration file "hyperref.cfg":

```
\providecommand*\Hy@defaultdriver}{hdvips}
for dvips, or
\providecommand*\Hy@defaultdriver}{hypertex}
for the default behaviour of hyperref.
```

See also the new option 'driverfallback'.

Backref entries

Alternative interface for formatting of backref entries, example:

```
\documentclass[12pt,UKenglish]{article}

\usepackage{babel}
\usepackage[pagebackref]{hyperref}

% Some language options are detected by package backref.
% This affects the following macros:
% \backrefpagesname
% \backrefsectionsname
% \backrefsep
% \backreftwosep
% \backreflastsep

\renewcommand*\backref[1]{
  % default interface
  % #1: backref list
  %
  % We want to use the alternative interface,
  % therefore the definition is empty here.
}

\renewcommand*\backrefalt[4]{%
  % alternative interface
  % #1: number of distinct back references
  % #2: backref list with distinct entries
  % #3: number of back references including duplicates
  % #4: backref list including duplicates
  \par
  #3 citation(s) on #1 page(s): #2,\par
  \ifnum#1=1 %
    \ifnum#3=1 %
      1 citation on page %
    \else
      #3 citations on page %
    \fi
  \else
    #3 citations on #1 pages %
  \fi
  #2,\par
  \ifnum#3=1 %
    1 citation located at page %
  \else
    #3 citations located at pages %
  \fi
}
```

```

    #4.\par
}

% The list of distinct entries can be further refined:
\renewcommand*{\backrefentrycount}[2]{%
  % #1: the original backref entry
  % #2: the count of citations of this entry,
  %    in case of duplicates greater than one
  #1%
  \ifnum#2>1 %
    ~(#2)%
  \fi
}

```

```

\begin{document}

  \section{Hello}
  \cite{ref1, ref2, ref3, ref4}
  \section{World}
  \cite{ref1, ref3}
  \newpage

  \section{Next section}
  \cite{ref1}
  \newpage

  \section{Last section}
  \cite{ref1, ref2}
  \newpage

  \pdfbookmark[1]{Bibliography}{bib}
  \begin{thebibliography}{99}
    \bibitem{ref1} Dummy entry one.

    \bibitem{ref2} Dummy entry two.

    \bibitem{ref3} Dummy entry three.

    \bibitem{ref4} Dummy entry four.

  \end{thebibliography}

\end{document}

```

\phantomsection

Set an anchor at this location. It is often used in conjunction with `\addcontentsline` for sectionlike things (index, bibliography, preface). `\addcontentsline` refers to the latest previous location where an anchor is set.

```

\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{\indexname}
\printindex

```

Now the entry in the table of contents (and bookmarks) for the index points to the start of the index page, not to a location before this page.

\hypercalcbp

See manual.

6 Package Compatibility

Currently only package loading orders are available:

Note: hyperref loads package "nameref" at `\begin{document}`. Sometimes this is too late, thus this package must be loaded earlier.

algorithm

```
\usepackage{float}
\usepackage{hyperref}
\usepackage[chapter]{algorithm}% eg.
```

amsmath

The environments `equation` and `eqnarray` are not supported too well. For example, there might be spacing problems (`eqnarray` isn't recommended anyway, see CTAN:info/l2tabu/, the situation for `equation` is unclear, because nobody is interested in investigating). Consider using the environments that package `amsmath` provide, e.g. `gather` for `equation`. The environment `equation` can even be redefined to use `gather`:

```
\usepackage{amsmath}
\let\equation\gather
\let\endequation\endgather
```

amsrefs

Package loading order:

```
\usepackage{hyperref}
\usepackage{amsrefs}
```

arydshln, longtable

Package `longtable` must be put before `hyperref` and `arydshln`, `hyperref` after `arydshln` generates an error, thus the resulting package order is then:

```
\usepackage{longtable}
\usepackage{hyperref}
\usepackage{arydshln}
```


babel/magyar.ldf

The old version 2005/03/30 v1.4j will not work.
 You need at least version 1.5, maintained by P\’eter Szab\’o,
 see CTAN:language/hungarian/babel/.

babel/spanish.ldf

Babel’s spanish.ldf redefines ‘\.’ to support ‘\...’.
 In bookmarks (\pdfstringdef) only ‘\.’ is supported.
 If ‘\...’ is needed,
`\texorpdfstring{\...}{\dots}`
 can be used instead.

bibentry

Workaround:

```
\makeatletter
\let\saved@bibitem\@bibitem
\makeatother

\usepackage{bibentry}
\usepackage{hyperref}

\begin{document}

\begingroup
  \makeatletter
  \let\@bibitem\saved@bibitem
  \nobibliography{database}
\endgroup
```

bigfoot

Hyperref does not support package ‘bigfoot’. And package
 ‘bigfoot’ does not support hyperref’s footnotes and disables
 them (hyperfootnotes=false).

chappg

Package ‘chappg’ uses \@addtoreset that is redefined by ‘hyperref’.
 The package order is therefore:

```
\usepackage{hyperref}
\usepackage{chappg}
```

cite

This is from Mike Shell:

```
> cite.sty cannot currently be used with hyperref.
> However, I can do a workaround via:
>
> \makeatletter
> \def\NAT@parse{\typeout{This is a fake Natbib command to fool Hyperref.}}
> \makeatother
```

```
>
> \usepackage[hypertex]{hyperref}
>
> so that hyperref will not redefine any of the biblabel stuff - so cite.sty
> will work as normal - although the citations will not be hyperlinked, of
> course (But this may not be an issue for many people).
```

countlto

Package ‘countlto’ adds several \@addtoreset commands that confuse ‘hyperref’. Therefore \theH<...> has to be fixed:

```
\usepackage{countlto}
\AtBeginDocument{% *after* \usepackage{countlto}
  \renewcommand*{\theHsection}{\theHchapter.\arabic{section}}%
  \renewcommand*{\theHsubsection}{\theHsection.\arabic{subsection}}%
  \renewcommand*{\theHsubsubsection}{\theHsubsection.\arabic{subsubsection}}%
  \renewcommand*{\theHparagraph}{\theHsubsubsection.\arabic{paragraph}}%
  \renewcommand*{\theHsubparagraph}{\theHparagraph.\arabic{subparagraph}}%
}
```

dblacnt

pd1enc.def or puenc.def should be loaded before:

```
\usepackage{hyperref}
\usepackage{dblacnt}
or see entry for "vietnam".
```

easyeqn

Not compatible, breaks.

ellipsis

This packages redefines \textellipsis, thus it has to be loaded after package hyperref (pd1enc.def/puenc.def should be loaded before):

```
\usepackage{hyperref}
\usepackage{ellipsis}
```

float

```
\usepackage{float}
\usepackage{hyperref}
```

- * Several \caption commands are not supported inside one float object.
- * Anchor are set at top of the float object, if its style is controlled by float.sty.

endnotes

Unsupported.

foiltex

Update to version 2008/01/28 v2.1.4b:

Since version 6.77a hyperref does not hack into \@beginndvi,

it uses package ‘atbegshi’ instead, that hooks into \shipout. Thus the patch of ‘foils.cls’ regarding hyperref is now obsolete and causes an undefined error message about \@hyperfixhead. This is fixed in FoilTeX 2.1.4b.

footnote

This package is not supported, you have to disable hyperref’s footnote support by using option “hyperfootnotes=false”.

geometry

Driver ‘dvipdfm’ and program ‘dvipdfm’ might generate a warning:

Sorry. Too late to change page size

Then prefer the program ‘dvipdfmx’ or use one of the following workarounds to move the \special of geometry to an earlier location:

```
\documentclass[dvipdfm]{article}% or other classes
\usepackage{atbegshi}
\AtBeginDocument{%
  \let\OrgAtBeginDvi\AtBeginDvi
  \let\AtBeginDvi\AtBeginShipoutFirst
}
\usepackage[
  paperwidth=170mm,
  paperheight=240mm
]{geometry}
\AtBeginDocument{%
  \let\AtBeginDvi\OrgAtBeginDvi
}
\usepackage{hyperref}
```

or

```
\documentclass[dvipdfm]{article}% or other classes
\usepackage{atbegshi}
\let\AtBeginDvi\AtBeginShipoutFirst
\usepackage[
  paperwidth=170mm,
  paperheight=240mm
]{geometry}
\usepackage{hyperref}
```

IEEEtran.cls

version >= V1.6b (because of \@makecaption, see ChangeLog)

index

version >= 1995/09/28 v4.1 (because of \addcontentsline redefinition)

lastpage

Compatible.

linguex

```
\usepackage{hyperref}
\usepackage{linguex}
```

ltabptch

```
\usepackage{longtable}
\usepackage{ltabptch}
\usepackage{hyperref}
```

mathenv

Unsupported.

Both ‘mathenv’ and ‘hyperref’ messes around with environment ‘eqnarray’. You can load ‘mathenv’ after ‘hyperref’ to avoid an error message. But \label will not work inside environment ‘eqnarray’ properly, for example.

minitoc-hyper

This package is obsolete, use the uptodate original package minitoc instead.

multind

```
\usepackage{multind}
\usepackage{hyperref}
```

natbib

```
\usepackage{natbib}
\usepackage{hyperref}
```

nomenc

* Example for introducing links for the page numbers:

```
\renewcommand*{\pagedeclaration}[1]{\unskip, \hyperpage{#1}}
```

* For equations the following might work:

```
\renewcommand*{\eqdeclaration}[1]{%
  \hyperlink{equation.#1}{(Equation~#1)}%
}
```

But the mapping from the equation number to the anchor name is not available in general.

prettyref

%%% example for prettyref %%%

```
\documentclass{article}
\usepackage{prettyref}
\usepackage[pdftex]{hyperref}
```

```
%\newrefformat{FIG}{Figure~\ref{#1}}% without hyperref
\newrefformat{FIG}{\hyperref[#{1}]{Figure~\ref*{#1}}}
```

```

\begin{document}
  This is a reference to \prettyref{FIG:ONE}.
  \newpage
  \begin{figure}
    \caption{This is my figure}
    \label{FIG:ONE}
  \end{figure}
\end{document}
%%% example for prettyref %%%

```

nttheorem

nttheorem-hyper.sty is an old patched version of ntheorem.sty.
 Newer versions of ntheorem know the option hyperref:
`\usepackage{hyperref}`
`\usepackage[hyperref]{nttheorem}`
 But there are still unsolved problems (options thref, ...).

setspace

```

\usepackage{setspace}
\usepackage{hyperref}

```

sidecap

Before 2002/05/24 v1.5h:
`\usepackage{nameref}`
`\usepackage{hyperref}`
`\usepackage{sidecap}`

subfigure

1995/03/06 v2.0:
`\usepackage{subfigure}`
`\usepackage{hyperref}`
 % hypertextnames is set to false.
v2.1:
`\usepackage{nameref}`
`\usepackage{subfigure}`
`\usepackage{hyperref}`
 or
`\usepackage{hyperref}`
`\usepackage{subfigure}`
v2.1.2:
 please update
v2.1.3:
`\usepackage{hyperref}`
`\usepackage{subfigure}`
 or vice versa?

titleref

```

\usepackage{nameref}
\usepackage{titleref}% without usetoc
\usepackage{hyperref}

```

tabularx

Linked footnotes are not supported inside environment ‘tabularx’, because they uses the optional argument of `\footnotetext`, see section ‘Limitations’. Before version 2011/09/28 6.82i `hyperref` had disabled footnotes entirely by ‘`hyperfootnotes=false`’.

titlesec

“`nameref`” supports `titlesec`, but `hyperref` does not (unsolved is the anchor setting, missing with unnumbered section, perhaps problems with page breaks with numbered ones).

ucs/utf8x.def

The first time a multibyte UTF8 sequence is called, it does some calculations and stores the result in a macro for speeding up the next calls of that UTF8 sequence. However this makes the first call non-expandable and will break if used in information entries or bookmarks. Package “`ucs`” offers `\PrerenderUnicode` or `\PreloadUnicodePage` to solve this:

```
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\usepackage{hyperref}% or with option unicode
\PrerenderUnicode{^^c3^^b6}% or \PrerenderUnicodePage{1}
\hypersetup{pdftitle={Umlaut example: ^^c3^^b6}}
```

The notation with two carets avoids trouble with 8-bit bytes for the README file, you can use the characters directly.

varioref

There are too many problems with `varioref`. Nobody has time to sort them out. Therefore this package is now unsupported.

Perhaps you are lucky and some of the features of `varioref` works with the following loading order:

```
\usepackage{nameref}
\usepackage{varioref}
\usepackage{hyperref}
```

Also some `babel` versions can be problematic. For example, 2005/05/21 v3.8g contains a patch for `varioref` that breaks the `hyperref` support for `varioref`.

Also unsupported:

- * `\Ref`, `\Vref` do not uppercase the first letter.
- * `\vpageref[...]`
On the same page a previous space is not suppressed.

verse

Version 2005/08/22 v2.22 contains support for `hyperref`.

For older versions see example from `de.comp.text.tex` (2005/08/11, slightly modified):

```

\documentclass{article}

% package order does not matter
\usepackage{verse}
\usepackage{hyperref}

\makeatletter
% make unique poemline anchors
\newcounter{verse@env}
\setcounter{verse@env}{0}
\let\org@verse\verse
\def\verse{%
  \stepcounter{verse@env}%
  \org@verse
}
\def\theHpoemline{\arabic{verse@env}.\thepoemline}

% add anchor for before \addcontentsline in \@vsptitle
\let\org@vsptitle\@vsptitle
\def\@vsptitle{%
  \phantomsection
  \org@vsptitle
}
\makeatother

\begin{document}

\poemtitle{Poem 1}
\begin{verse}
An one-liner.
\end{verse}

\newpage

\poemtitle{Poem 2}
\begin{verse}
Another one-liner.
\end{verse}

\end{document}

```

vietnam

```

% pdlenc.def should be loaded before package dblacnt:
\usepackage[PD1,OT1]{fontenc}
\usepackage{vietnam}
\usepackage{hyperref}

```

XeTeX

Default for the encoding of bookmarks is ‘pdfencoding=auto’. That means the strings are always treated as unicode strings. Only if the string restricts to the printable ASCII set, it is written as ASCII string. The reason is that the `\special` does not support PDFDocEncoding.

XeTeX uses the program `xdvipdfmx` for PDF output generation. This program behaves a little different from `dvipdfm`, because of the supported Unicode characters. Strings for bookmarks or information entries can be output directly. The big chars (char code > 255) are written in UTF-8 and `xdvipdfmx` tries to convert them to UTF-16BE. However `hyperref` already provides PDF strings encoded in UTF-16BE, thus the result is a warning

"Failed to convert input string to UTF16..."

The best way would be, if `xdvipdfm` could detect the byte order marker (`\376\377`) and skips the conversion if that marker is present.

For the time being I added the following to `hyperref`, when option `'pdfencoding=auto'` is set (default for XeTeX): The string is converted back to big characters thus that the string is written as UTF-8. But I am very unhappy with this solution. Main disadvantage:

Two versions of `\pdfstringdef` are needed:

- a) The string is converted back to big characters for the "tainted keys" of `xdvipdfmx` (`spc_pdfm.c: default_taintkeys`). The subset `hyperref` uses is `/Title`, `/Author`, `/Subject`, `/Keywords`, `/Creator`, `/Producer`, `/T`. Any changes of this set in `xdvipdfmx` cannot be detected by `hyperref`.
- b) Without conversion for the other strings, providing UTF16be directly. Examples: Prefix of page labels, some elements of formulars.

Thus *each* application that uses `\pdfstringdef` now must check, if it defines a string for some of the tainted keys.

If yes, then the call of `\pdfstringdef` should be preceded by `"\csname HyPsd@XeTeXBigCharstrue\endcsname"`.

Example: package `bookmark`.

7 Limitations

Wrapped/broken link support

Only few drivers support automatically wrapped/broken links, e.g. `pdftex`, `dvipdfm`, `hypertex`. Other drivers lack this feature, e.g. `dvips`, `dvipsone`.

Workarounds:

- * For long section or caption titles in the table of contents or list of figures/tables option `"linktocpage"` can be used. Then the page number will be a link, and the overlong section title is not forced into an one line link with overvull `\hbox` warning.
- * `"\url"`s are caught by package `"breakurl"`.
- * The option `"breaklinks"` is intended for internal use. But it can be used to force link wrapping, e.g. when printing a document. However, when such a document is converted to PDF and viewed with a PDF viewer, the active link area will be misplaced.

Another limitation: some penalties are "optimized" by TeX, thus there are missing break points, especially within `\url`. (See thread `"hyperref.sty, breaklinks and url.sty 3.2"`)

in comp.text.tex 2005-09).

Links across pages

In general they have problems:

- * Some driver doesn't support them at all (see above).
- * The driver allows it, but the link result might include the footer and/or header, or an error message can occur sometimes.

Footnotes

LaTeX allows the separation of the footnote mark and the footnote text (`\footnotemark`, `\footnotetext`). This interface might be enough for visual typesetting. But the relation between `\footnotemark` to `\footnotetext` is not as strong as `\ref` to `\label`. Therefore it is not clear in general which `\footnotemark` references which `\footnotetext`. But that is necessary to implement hyperlinking. Thus the implementation of `hyperref` does not support the optional argument of `\footnotemark` and `\footnotetext`.

8 Hints

Spaces in option values

Unhappily LaTeX strips spaces from options if they are given in `\documentclass` or `\usepackage` (or `\RequirePackage`), e.g.:

```
\usepackage[pdfborder=0 0 1]{hyperref}
```

Package `hyperref` now gets

```
pdfborder=001
```

and the result is an invalid PDF file.

As workaround braces can be used:

```
\usepackage[pdfborder={0 0 1}]{hyperref}
```

Some options can also be given in `\hypersetup`:

```
\hypersetup{pdfborder=0 0 1}
```

In `\hypersetup` the options are directly processed as key value options (see package `keyval`) without space stripping in the value part.

Alternatively, LaTeX's option handling system can be adapted to key value options by one of the packages "`kvoptions-patch`" (from project "`kvoptions`") or "`xkvltxp`" (from project "`xsetkeys`").

Index with makeindex

- * Package `hyperref` adds `\hyperpage` commands by the `encap` mechanism (see documentation of `Makeindex`), if option `hyperindex` is set (default). `\hyperpage` uses the page anchors that are set by `hyperref` at each page (default). However in the default case page numbers are used in anchor names in arabic form. If the page numbers in other formats are used (book class with `\frontmatter`, `\romannumbering`, ...), then the page anchors are not unique. Therefore option "`plainpages=false`" is recommended.
- * The `encap` mechanism of `Makeindex` allows to use one command only

(see documentation of Makeindex).

If the user sets such a command, hyperref suppresses its `\hyperpage` command. With logical markup this situation can easily be solved:

```
\usepackage{makeidx}
\makeindex
\usepackage[hyperindex]{hyperref}
\newcommand*{\main}[1]{\textbf{\hyperpage{#1}}}
```

...

```
\index{Some example|main}
```

- * Scientific Word/Scientific WorkPlace users can use package `robustindex` with `hyperindex=false`.
- * Other encap characters can be set by option "encap".

Example for use of "?":

```
\usepackage[encap=?]{hyperref}
```

- * An other possibility is the insertion of `\hyperpage` by a style file for `makeindex`. For this case, `hyperref`'s insertion will be disabled by "`hyperindex=false`". `\hyperpage` will be defined regardless of setting of `hyperindex`.

```
%%% cut %%% hyperindex.ist %%% cut %%%
delim_0 ", \hyperpage{"
delim_1 ", \hyperpage{"
delim_2 ", \hyperpage{"
delim_n "}", \hyperpage{"
delim_t "}"
encap_prefix "}"
encap_infix "{\hyperpage{"
encap_suffix "}"
%%% cut %%% hyperindex.ist %%% cut %%%
```

Warning "bookmark level for unknown <foobar> defaults to 0"

Getting rid of it:

```
\makeatletter
\providecommand*\toclevel@<foobar>{0}
\makeatother
```

Link anchors in figures

The caption command increments the counter and here is the place where `hyperref` set the corresponding anchor. Unhappily the caption is set below the figure, so the figure is not visible if a link jumps to a figure.

In this cases, try package "`hypcap.sty`" that implements a method to circumvent the problem.

Additional unicode characters in bookmarks and pdf information entries:

```
\documentclass[pdftex]{article}
\usepackage[unicode]{hyperref}
```

```
% Support for additional unicode characters:
%
% Example: \.{a} and \d{a}
```

```

%
% 1. Get a list with unicode data, eg:
%   http://www.unicode.org/Public/UNIDATA/UnicodeData.txt
%
% 2. Identify the characters (\{a}, \d{a}):
%
%   0227;LATIN SMALL LETTER A WITH DOT ABOVE;...
%   1EA1;LATIN SMALL LETTER A WITH DOT BELOW;...
%
% 3. Calculate the octal code:
%   The first characters of the line in the file are
%   hex values, convert each byte and prepend them
%   with a backslash. (This will go into the PDF file.)
%
%   0227 -> \002\047
%   1EA1 -> \036\241
%
% 4. Transform into a form understood by hyperref:
%
%   Hyperref must know where the first byte starts,
%   this is marked by "9" (8 and 9 cannot occur in
%   octal numbers):
%
%   \002\047 -> \9002\047
%   \036\241 -> \9036\241
%
%   Optional: "8" is used for abbreviations:
%   \900 = \80, \901 = \81, \902 = \82, ...
%
%   \9002\047 -> \82\047
%
% 5. Declare the character with LaTeX:
%
\DeclareTextCompositeCommand{\.}{PU}{a}{\82\047}
\DeclareTextCompositeCommand{\d}{PU}{a}{\9036\241}

\begin{document}
\section{\={a}, \d{a}, \'{a}, \.a}
\end{document}

```

Footnotes

The footnote support is rather limited. It is beyond the scope to use `\footnotemark` and `\footnotetext` out of order or reusing `\footnotemark`. Here you can either disable `hyperref`'s footnote support by `"hyperfootnotes=false"` or fiddle with internal macros, nasty examples:

```

\documentclass{article}
\usepackage{hyperref}
\begin{document}
Hello%
\footnote{The first footnote}
World%
\addtocounter{footnote}{-1}%

```

```

\addtocounter{Hfootnote}{-1}%
\footnotemark.
\end{document}

or

\documentclass{article}

\usepackage{hyperref}

\begin{document}

\makeatletter

A%
\footnotemark
\let\saved@Href@A\Hy@footnote@currentHref
% remember link name
B%
\footnotemark
\let\saved@Href@B\Hy@footnote@currentHref
b%
\addtocounter{footnote}{-1}%
\addtocounter{Hfootnote}{-1}% generate the same anchor
\footnotemark
C%
\footnotemark
\let\saved@Href@C\Hy@footnote@currentHref

\addtocounter{footnote}{-2}%
\let\Hy@footnote@currentHref\saved@Href@A
\footnotetext{AAAA}%
\addtocounter{footnote}{1}%
\let\Hy@footnote@currentHref\saved@Href@B
\footnotetext{BBBBB}%
\addtocounter{footnote}{1}%
\let\Hy@footnote@currentHref\saved@Href@C
\footnotetext{CCCC}%

\end{document}

```

Subordinate counters

Some counters do not have unique values and require the value of other counters to be unique. For example, sections or figures might be numbered within chapters or `\newtheorem` is used with an optional counter argument. Internally LaTeX uses `\@addtoreset` to reset a counter in dependency to another counter. Package `hyperref` hooks into `\@addtoreset` to catch this situation. Also `\numberwithin` of package `amsmath` is caught by `hyperref`.

However, if the definition of subordinate counters take place before `hyperref` is loaded, the old meaning of `\@addtoreset` is called without `hyperref`'s additions. Then the companion counter macro `\theH<counter>` can be redefined accordingly. Or move the definition of subordinate counters after `hyperref` is loaded.

Example for `\newtheorem`, problematic case:

```
\newtheorem{corA}{CorollaryA}[section]
\usepackage{hyperref}
```

Solution a)

```
\usepackage{hyperref}
\newtheorem{corA}{CorollaryA}[section]
```

Solution b)

```
\newtheorem{corA}{CorollaryA}[section]
\usepackage{hyperref}
\newcommand*{\theHcorA}{\theHsection.\number\value{corA}}
```

9 Authors/Maintainers

- * Sebastian Rahtz
- * Heiko Oberdiek

10 Bug Reports

A bug report should contain:

- * Comprehensive problem description. This includes error or warning messages.
 - * `\errorcontextlines=\maxdimen` can be added in the TeX code to get more informations in TeX error messages.
- * Minimal test file that shows the problem, but does not contain any unnecessary packages and code.
- * Used drivers/programs.
- * Version information about used packages and programs.
 - * If you are using LaTeX, then add `"\listfiles"`. Then a list of version informations is printed at the end of the LaTeX run.
- * Please no other files than the minimal test file.

The other files `.log`, `.dvi`, `.ps`, `.pdf` are seldom necessary, so send them only on request.

Bug address

A bug tracker is available at Puszczka:

<https://puszczka.gnu.org.ua/bugs/?group=hyperref>

Alternatively bug reports can be send to the maintainer:

Heiko Oberdiek <heiko [dot] oberdiek [at] googlemail [dot] com>

Vietnamese part

Responsible for the Vietnamese translations of the

`\autoref` names and `puvnenc.def` are:

Han The Thanh <hanthethanh at gmail.com>

Reinhard Kotucha <reinhard.kotucha at web.de>

Arabic part

Responsible for the additions to PU encoding for Arabi is

Youssef Jabri <yjabri@ensa.univ-oujda.ac.ma>

11 Known Problems

- * (half-done) hyper images (link from thumbnail in text)
- * Relative links are not sorted out or documented well.
For PDF generation:
 - * With baseurl: all links are considered relative to this URL.
 - * Without baseurl: a relative link without "file:" can be achieved by:


```
\begingroup
  \hypersetup{linkfileprefix={}}%
  \href{../foo/bar.html}{bar.html}
\endgroup
```
- * ...

12 ToDo

- * modules
- * bookmark organisation
- * documentation
- * PDF threads
- * more for PDF forms
 - * per object setting
 - * vary gap between text and box
- * PostScript driver: the current implementation doesn't rely support nested links. The start positions should be remembered in a stack, but there are complications with page breaks.
- * ...

13 Versions in TeX distributions

TeX Live

- * TL 2011: 2011/04/17 v6.82g (at time of first release)
- * TL 2010: 2010/06/18 v6.81g (at time of first release)
- * TL 2009: 2009/10/09 v6.79a (at time of first release)
- * TL 2008: 2008/08/14 v6.78f (at time of first release)
- * TL 2007: 2007/02/07 v6.75r
- * TL 2005: 2003/11/30 v6.74m
- * TL 2004: 2003/11/30 v6.74m
- * TL 2003: 2003/09/15 v6.74i
- * TL 7 (2002): 2002/05/27 v6.72r
- * TL 6b (2001): 2001/05/26 v6.71g
- * TL 5d (2000): 2000/07/02 v6.70m
- * TL 5c (2000): 2000/05/08 v6.70f
- * TL 4 (1999): 1999/04/13 v6.56
- * TL 3 (1998): 1998/03/25 v6.19