Hypertext marks in LaTeX: a manual for hyperref

Sebastian Rahtz* Heiko Oberdiek† The LaTeX3 Project‡

2023-10-21 v7.01c

Contents

*deceased
†inactive
‡https://github.com/latex3/hyperref/issues

1 Preface

As can be already seen in the following introduction, hyperref has a long history and has seen many changes over time. The introduction mentions workflows, drivers and problems that are no longer (or only in edge cases) relevant. The documentation reflects this varied history: changes and extensions and explanations were and are spread over various papers and sources or have been incorporated later and so are not always in a coherent order and in sync which each other.

This history is continuing: If you are using the new LaTeX PDF management which is currently distributed as a testphase package pdfmanagement-testphase then hyperref will for the PDF output use a new generic driver which contains a number of changes and new features. The documentation of this driver hyperref-generic.pdf is currently a part of the pdfmanagement-testphase documentation. One important change of the new driver is that it removed the old hyperref code for book marks and uses the bookmark package instead. So to learn about options to extend the bookmarks you should consult the bookmark documentation too.

1.1 Restoring removed patches

hyperref has over time patched quite a number of packages to resolve clashes and incompabilities. Many of them are either no longer needed or should be done by the original packages. Those patches are now slowly removed from hyperref. It should normally not lead to problems, but in case that the patches should be restored they can be loaded through the package hyperref-patches which is a part of this bundle.

2 Introduction

The package derives from, and builds on, the work of the HyperTeX project, described at `http://xxx.lanl.gov/hypertex/`[1]. It extends the functionality of all the LaTeX cross-referencing commands (including the table of contents, bibliographies etc) to produce `\special` commands which a driver can turn into hypertext links; it also provides new commands to allow the user to write *ad hoc* hypertext links, including those to external documents and URLs.

The package is currently maintained at https://github.com/latex3/hyperref/ and issues should be reported there.

This manual provides a brief overview of the hyperref package. For more details, you should read the additional documentation distributed with the package, as well as the complete documentation by processing `hyperref.dtx`. You should also read the chapter on hyperref in *The LaTeX Web Companion*, where you will find additional examples.

The HyperTeX specification[2] says that conformant viewers/translators must recognize the following set of `\special` constructs:

href: `html:<a href = "href_string">`

name: `html:<a name = "name_string">`

end: `html:</a>`

image: `html:<img src = "href_string">`

base_name: `html:<base href = "href_string">`

The *href*, *name* and *end* commands are used to do the basic hypertext operations of establishing links between sections of documents. The *image* command is intended (as with current HTML viewers) to place an image of arbitrary graphical format on the page in the current location. The *base_name* command is be used to communicate to the DVI viewer the full (URL) location of the current document so that files specified by relative URLs may be retrieved correctly.

The *href* and *name* commands must be paired with an *end* command later in the TeX file—the TeX commands between the two ends of a pair form an *anchor* in the document. In the case of an *href* command, the *anchor* is to be highlighted in the *DVI viewer*, and when clicked on will cause the scene to shift to the destination specified by *href_string*. The *anchor* associated with a

[1] Now: https://ctan.org/tex-archive/support/hypertex/hypertex

[2] This is borrowed from an article by Arthur Smith.

name command represents a possible location to which other hypertext links may refer, either as local references (of the form `href="#name_string"` with the *name_string* identical to the one in the name command) or as part of a URL (of the form *URL#name_string*). Here *href_string* is a valid URL or local identifier, while *name_string* could be any string at all: the only caveat is that '"' characters should be escaped with a backslash (\), and if it looks like a URL name it may cause problems.

However, the drivers intended to produce *only* PDF use literal PostScript or PDF \special commands. The commands are defined in configuration files for different drivers, selected by package options or for most current engines autodetected; at present, the following drivers are supported:

hypertex DVI processors conforming to the HyperTEX guidelines (i.e. xdvi, dvips (with the -z option), OzTEX, and Textures)

dvips produces \special commands tailored for dvips. This is the default driver if dvi mode is detected.

dvipsone produces \special commands tailored for dvipsone

ps2pdf a special case of output suitable for processing by earlier versions of Ghostscript's PDF writer; this is basically the same as that for dvips, but a few variations remained before version 5.21

tex4ht produces \special commands for use with TEX4ht, autodetected.

pdftex pdfTEX, Hàn Thế Thành's TEX variant that writes PDF directly, autodetected.

luatex luaTEX, Unicode TEX variant that writes PDF directly, autodetected.

dvipdfm produces \special commands for Mark Wicks' DVI to PDF driver dvipdfm

dvipdfmx produces \special commands for driver dvipdfmx, a successor of dvipdfm

dviwindo produces \special commands that Y&Y's Windows previewer interprets as hypertext jumps within the previewer

vtex produces \special commands that MicroPress' HTML and PDF-producing TEX variants interpret as hypertext jumps within the previewer, autodetected.

textures produces \special commands that Textures interprets as hypertext jumps within the previewer

xetex produces \special commands for XeTEX, autodetected.

hitex (new 2023) for the hint format produces by the HiTEX engine, autodetected. This a quite experimental engine and the driver file is not part of the hyperref bundle but is provided by the hitex package. Problems should be reported to https://github.com/ruckertm/HINT.

Output from dvips or dvipsone must be processed using Acrobat Distiller to obtain a PDF file.[3] The result is generally preferable to that produced by using the hypertex driver, and then processing with dvips -z, but the DVI file is not portable. The main advantage of using the HyperTEX \special commands is that you can also use the document in hypertext DVI viewers, such as xdvi.

[3] Make sure you turn off the partial font downloading supported by dvips and dvipsone in favor of Distiller's own system.

driverfallback If a driver is not given and cannot be autodetected, then use the driver option, given as value to this option driverfallback. Example:

```
driverfallback=dvipdfm
```

Autodetected drivers (pdftex, luatex, xetex, vtex, vtexpdfmark, hitex) are recognized from within TEX and therefore cannot be given as value to option driverfallback. However a DVI driver program is run after the TEX run is finished. Thus it cannot be detected at TEX macro level. Then package hyperref uses the driver, given by driverfallback. If the driver is already specified or can be autodetected, then option driverfallback is ignored.

3 Implicit behavior

This package can be used with more or less any normal LATEX document by specifying in the document preamble

```
\usepackage{hyperref}
```

Make sure it comes *last* of your loaded packages, to give it a fighting chance of not being over-written, since its job is to redefine many LATEX commands.[4]

Do not load it in `\AtBeginDocument` *or the* `begindocument` *hook!* While this often worked in the past this is not officially supported. As hyperref and nameref use this hook too to initialize commands, timing of code execution is tricky and fragile if the packages are loaded there. If you want to delay the loading, use the `begindocument/before` hook.

Hopefully you will find that all cross-references work correctly as hypertext. For example, `\section` commands will produce a bookmark and a link, whereas `\section*` commands will only show links when paired with a corresponding `\addcontentsline` command.

In addition, the `hyperindex` option (see below) attempts to make items in the index by hyperlinked back to the text, and the option `backref` inserts extra 'back' links into the bibliography for each entry. Other options control the appearance of links, and give extra control over PDF output. For example, `colorlinks`, as its name well implies, colors the links instead of using boxes; this is the option used in this document.

4 Interfaces for class and package authors

hyperlink features are nowadays a common requirement. hyperref patches quite a number of commands from the LATEX core and from packages to add such features. But this is rather fragile and it add dependencies on the loading order and can break if the external packages changes. It is therefore much better if packages add suitable support to their commands directly. Quite a lot packages actually did this, but due to missing documentation of the interface they often looked into the code and then used internal commands not meant as public command.

The following tries to describe the existing variables and commands that can be viewed as public interfaces or which should or can be set by packages to stay compatible with the hyperref command. Documented user commands are naturally interfaces too, they are not explicitly mentioned here again.

This section is work in progress. Suggestions or comments are welcome.

[4] But work has started to reduce the number of redefinition and so the dependencies on the loading order.

4.1 Counters

Counters play an important part in the code. They are used to create destination names and to define hierarchies like the bookmarks. To work correctly often they require some additional setups.

\theH<counter> hyperref creates destination names for link anchor typically out of the name of the counter and the \the<counter> value. This can fail, e.g. if \the<counter> is not unique through the document, or if it is not expandable. In such cases \theH<counter> should be defined so that it gives a unique, expandable value. It doesn't harm to define it even if hyperref is not loaded.

\toclevel@<counter> This is a variable which should contain a number. It is used for the level in the bookmarks. It should be defined for all counters which are used in toc like lists and \addcontentsline. Typical values are

```
\def\toclevel@part{-1}
\def\toclevel@chapter{0}
\def\toclevel@section{1}
\def\toclevel@subsection{2}
\def\toclevel@subsubsection{3}
\def\toclevel@paragraph{4}
\def\toclevel@subparagraph{5}
\def\toclevel@figure{0}
```

4.2 Values of package and \hypersetup options

When a key is set either in the package options or with \hypersetup hyperref typically stores the result in internal variables, or executes some code or sets a internal boolean. Package and class authors should here *not* rely on the names or the details of the key processing. New in version 7.00s

But as other packages sometimes need to know which value has been set, some values can be retrieved with the expandable \GetDocumentProperties. The values are given back surrounded by \exp_not:n, so can be used safely in an \edef. So for example to get the pdfauthor you can do.

```
\edef\mypdfauthor{\GetDocumentProperties{hyperref/pdfauthor}}
```

The values are given back as entered by the user! If they should be used in a PDF context \pdfstringdef or something equivalent must still be applied.

Currently this interface can be used for the keys pdfauthor, pdftitle, pdfproducer, pdfcreator, pdfsubject and pdfkeywords. If used with a unknown key an empty value is returned. The interface works also if the new PDF management is loaded with \DocumentMetadata, in this case more keys gives back their value.

4.3 Links commands

The following commands are provided by all drivers to create links. They can be used by packages if the user commands are not sufficient. New drivers must provide this commands with similar arguments.

```
\hyper@anchor {destination name}
\hyper@anchorstart {destination name}
\hyper@anchorend
```

```
\hyper@link   {context}{destination name}{link text}  %GoTo
\hyper@linkstart {context} {destination name}        %GoTo
\hyper@linkend                                  %GoTo
\hyper@linkfile  {link text} {filename} {destname}    %GoToR
\hyper@linkurl   {link text}{url}                  %URI
\hyper@linklaunch{filename} {link text} {Parameters}  %Launch, only with new generic driver
\hyper@linknamed {action}{link text}                %Named, only with new generic driver
```

4.4 Creating targets

Internal links and bookmarks need something they can jump to. In a PDF this is normally called a *destination* (and the primitive is therefor called \pdfdest), in HTML it is more common to call this an *anchor* (and the hyperref uses therefor \hyper@anchor). History can not be undone but future commands and descriptions will use the generic *target* unless the PDF specific destination is meant.

Targets are created automatically when \refstepcounter is used and in many cases this does the right thing and nothing more is needed. But there are exceptions:

- A needed target can be missing for example if a sectioning command doesn't have a number as the starred version is used or due to the setting of secnumdepth.
- The target created by the \refstepcounter can be in the wrong place.
- The target created by the \refstepcounter can affect spacing.
- The target name created by the \refstepcounter is not usable, e.g. in \bibitem where you need a target name bases on the bib-key.

Package authors and users can use the following commands to create and manipulate targets. The commands are described in more detail in hyperref-linktarget.pdf.

```
\MakeLinkTarget
\LinkTargetOff
\LinkTargetOn
\NextLinkTarget
\SetLinkTargetFilter
```

The first four commands will be defined also in LaTeX directly as no-op and so can be used even if hyperref is not loaded.

Until LaTeX is updated package authors can also provide these definitions directly:

```
\ProvideDocumentCommand\MakeLinkTarget{sO{}m}{}
\ProvideDocumentCommand\LinkTargetOn{}{}
\ProvideDocumentCommand\LinkTargetOff{}{}
\ProvideDocumentCommand\NextLinkTarget{m}{}
```

4.5 Patches and how to suppress them

The patches to external commands made by hyperref can be avoided completely by loading hyperref with the option implicit=false. But suppressing everything is often too drastic. There is a work in progress to classify the patches and to offer interfaces to suppress them in a more granular way.

sectioning commands
- hyperref patches \@sect, \@ssect, \@chapter, \@schapter, \@part, \@spart.

- It adds to the starred commands a target for a link (with the prefix chapter* for chapters and section* otherwise). To the other commands it adds a target for a link if the sectioning is unnumbered, e.g. because of the secnumdepth setting or in the front matter.
- The patch can be suppressed by defining the command \hyper@nopatch@sectioning. This should normally be done only by a class or a package which provides sectioning commands and adds the targets itself. Targets have a location on the page and e.g. the section commands should take indents into account. Targets are needed for bookmarks and the table of contents, so \@currentHref should get the correct meaning before \addcontentsline is used.
- Note that the nameref package patches these commands too to add commands to store the title text in \@currentlabelname. Check the nameref documentation about a way to suppress these patches.

footnotes To enable (partly) the linking of footnotes hyperref redefines or patches various commands, in part package dependant.

- hyperref redefines \@xfootnotenext, \@xfootnotemark, \@mpfootnotetext, \@footnotetext, \@footnotemark. If tabularx is loaded it changes \TX@endtabularx. If longtable is loaded it changes \LT@p@ftntext. If fancyvfb is loaded it redefines \V@@footnotetext. It also redefines \footref and \maketitle.
- *All* those redefinitions can be suppressed by defining \hyper@nopatch@footnote. Be aware that this can suppress links but also make unwanted links appear.

amsmath tags hyperref redefines two internal commands of amsmath related the \tag command to add an anchor. This code can be suppressed by defining \hyper@nopatch@amsmath@tag. (This normally makes no sense in packages but will probably be needed when math environments are changed to allow tagging.)

counters hyperref patches the kernel command \@definecounter, \@newctr, \@addtoreset and the amsmath command \numberwithin to ensure that for every counter the correct \theHcounter representation is created or reset. This code can be suppressed by defining \hyper@nopatch@counter. (This normally makes no sense in packages but will probably be needed when kernel commands are changed to allow tagging.)

math environments hyperref patches \equation/\endequation, \eqnarray, \endeqnarray. This code can be suppressed by defining \hyper@nopatch@mathenv.

table of contents hyperref redefines \contentsline to be able to add links to toc entries. It redefines \addcontentsline to create the bookmarks and pass the destination names to the toc entries. This code can be suppressed by defining \hyper@nopatch@toc.

captions hyperref redefines \caption and \@caption to insert targets for links. This code can be suppressed by defining \hyper@nopatch@caption (additional helper commands are not suppressed). As various packages redefine captions too (e.g. the caption package) side-effects must be carefully tested!

longtable hyperref redefines \LT@start and \LT@array to move the targets in a better place. This code can be suppressed by defining \hyper@nopatch@longtable

theorems hyperref patches \@thm. This code can be suppressed by defining \hyper@nopatch@thm

citations and bibliography If natbib is not loaded hyperref redefines \bibcite, \@lbibitem and \@bibitem. These redefinitions can be suppressed by defining \hyper@nopatch@bib.

5 Package options

All user-configurable aspects of hyperref are set using a single 'key=value' scheme (using the keyval package) with the key Hyp. The options can be set either in the optional argument to the \usepackage command, or using the \hypersetup macro. When the package is loaded, a file hyperref.cfg is read if it can be found, and this is a convenient place to set options on a site-wide basis.

Note however that some options (for example unicode) can only be used as package options, and not in \hypersetup as the option settings are processed as the package is read. The following tabular lists (hopefully all) these options. Be aware that some of the option do nothing or have changed behaviour if the new pdfmanagement and so the new generic hyperref driver is used.

option	remark
all driver options, e.g. pdftex	often not needed, as detected automatically
implicit	
pdfa	no-op with new pdfmanagement, set the standard in \DeclareDocumentMetadata.
unicode	is the default now anyway
pdfversion	no-op with new pdfmanagement, set the version in \DeclareDocumentMetadata.
bookmarks	this will probably change at some time.
backref	
pagebackref	
destlabel	
pdfusetitle	
pdfpagelabels	
hyperfootnotes	
hyperfigures	
hyperindex	
encap	
CJKbookmarks	only with the new pdfmanagement, in other cases it can be used in \hypersetup
psdextra	only with the new pdfmanagement, in other cases it can be used in \hypersetup
nesting	only with the new pdfmanagement, in other cases it can be used in \hypersetup (but is quite unclear if it has any use)

As an example, the behavior of a particular file could be controlled by:

- a site-wide hyperref.cfg setting up the look of links, adding backreferencing, and setting a PDF display default:

```
\hypersetup{backref,
```

```
pdfpagemode=FullScreen,
colorlinks=true}
```

- A global option in the file, which is passed down to hyperref:

```
\documentclass[dvips]{article}
```

- File-specific options in the \usepackage commands, which override the ones set in hyperref.cfg:

```
\usepackage[colorlinks=false]{hyperref}
\hypersetup{pdftitle={A Perfect Day}}
```

As seen in the previous example, information entries (pdftitle, pdfauthor, …) should be set after the package is loaded. Otherwise LaTeX expands the values of these options prematurely. Also LaTeX strips spaces in options. Especially option 'pdfborder' requires some care. Curly braces protect the value, if given as package option. They are not necessary in \hypersetup.

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

Some options can be given at any time, but many are restricted: before \begin{document}, only in \usepackage[...]{hyperref}, before first use, etc.

In the key descriptions that follow, many options do not need a value, as they default to the value true if used. These are the ones classed as 'boolean'. The values true and false can always be specified, however.

5.1 General options

Firstly, the options to specify general behavior and page size.

draft	boolean	*false*	all hypertext options are turned off
final	boolean	*true*	all hypertext options are turned on
debug	boolean	*false*	extra diagnostic messages are printed in the log file
verbose	boolean	*false*	same as debug
implicit	boolean	*true*	redefines LaTeX internals
setpagesize	boolean	*true*	sets page size by special driver commands

5.2 Options for destination names

Destinations names (also anchor, target or link names) are internal names that identify a position on a page in the document. They are used in link targets for inner document links or the bookmarks, for example.

Usually anchor are set, if \refstepcounter is called. Thus there is a counter name and value. Both are used to construct the destination name. By default the counter value follows the counter name separated by a dot. Example for the fourth chapter:

```
chapter.4
```

This scheme is used by:

\autoref displays the description label for the reference depending on the counter name.

\hyperpage is used by the index to get page links. Page anchor setting (pageanchor) must not be turned off.

It is very important that the destination names are unique, because two destinations must not share the same name. The counter value \the<counter> is not always unique for the counter. For example, table and figures can be numbered inside the chapter without having the chapter number in their number. Therefore hyperref has introduced \theH<counter> that allows a unique counter value without messing up with the appearance of the counter number. For example, the number of the second table in the third chapter might be printed as 2, the result of \thetable. But the destination name table.2.4 is unique because it has used \theHtable that gives 2.4 in this case.

Often the user do not need to set \theH<counter>. Defaults for standard cases (chapter, …) are provided. And after hyperref is loaded, new counters with parent counters also define \theH<counter> automatically, if \newcounter, \@addtoreset or \numberwithin of package amsmath are used.

Usually problems with duplicate destination names can be solved by an appropriate definition of \theH<counter>. If option hypertexnames is disabled, then a unique artificial number is used instead of the counter value. In case of page anchors the absolute page anchor is used. With option plainpages the page anchors use the arabic form. In both latter cases \hyperpage for index links is affected and might not work properly.

If an unnumbered entity gets an anchor (starred forms of chapters, sections, …) or \phantomsection is used, then the dummy counter name section* and an artificial unique number is used.

If the final PDF file is going to be merged with another file, than the destination names might clash, because both documents might contain chapter.1 or page.1. Also hyperref sets anchor with name Doc-Start at the begin of the document. This can be resolved by redefining \HyperDestNameFilter. Package hyperref calls this macro each time, it uses a destination name. The macro must be expandable and expects the destination name as only argument. As example, the macro is redefined to add a prefix to all destination names:

```
\renewcommand*{\HyperDestNameFilter}[1]{\jobname-#1}
```

In document docA the destination name chapter.2 becomes docA-chapter.2.

Destination names can also be used from the outside in URIs(, if the driver has not removed or changed them), for example:

```
http://somewhere/path/file.pdf#nameddest=chapter.4
```

However using a number seems unhappy. If another chapter is added before, the number changes. But it is very difficult to pass a new name for the destination to the anchor setting process that is usually deep hidden in the internals. The first name of \label after the anchor setting seems a good approximation:

```
\section{Introduction}
\label{intro}
```

Option destlabel checks for each \label, if there is a new destination name active and replaces the destination name by the label name. Because the destination name is already in use because of the anchor setting, the new name is recorded in the .aux file and used in the subsequent LaTeX run. The renaming is done by a redefinition of \HyperDestNameFilter. That leaves the old destination names intact (e.g., they are needed for \autoref). This redefinition is also available as \HyperDestLabelReplace, thus that an own redefinition can use it. The following example also adds a prefix for *all* destination names:

```
\renewcommand*{\HyperDestNameFilter}[1]{%
  \jobname-\HyperDestLabelReplace{#1}%
}
```

The other case that only files prefixed that do not have a corresponding `\label` is more complicate, because `\HyperDestLabelReplace` needs the unmodified destination name as argument. This is solved by an expandable string test (`\pdfstrcmp` of pdfTeX or `\strcmp` of XeTeX, package pdftexcmds also supports LuaTeX):

```
\usepackage{pdftexcmds}
\makeatletter
\renewcommand*{\HyperDestNameFilter}[1]{%
  \ifcase\pdf@strcmp{#1}{\HyperDestLabelReplace{#1}} %
    \jobname-#1%
  \else
    \HyperDestLabelReplace{#1}%
  \fi
}
\makeatother
```

With option `destlabel` destinations can also named manually, if the destination is not yet renamed:

`\HyperDestRename{`⟨*destination*⟩`}{`⟨*newname*⟩`}`

Hint: Anchors can also be named and set by `\hypertarget`.

`destlabel`	boolean	*false*	destinations are named by first `\label` after anchor creation
`hypertexnames`	boolean	*true*	use guessable names for links
`naturalnames`	boolean	*false*	use LaTeX-computed names for links
`plainpages`	boolean	*false*	Forces page anchors to be named by the Arabic form of the page number, rather than the formatted form.

5.3 Page anchors

In a PDF hyperref adds to every page an target for links (in the upper left corner). These targets are for example used by the index to references the pages of the entries. Unlike the targets generated by other counters it is not possible to change the names of these targets by defining a `\theHpage` command: the name can currently not be stored in a label and so is not referenceable. Nevertheless a few options exists, they are already mentioned elsewhere in the documentation but we provide here a summary:

`pageanchor` A boolean option that determines whether every page is given an target at the top left corner. If this is turned off, `\printindex` will not contain valid hyperlinks.

`hypertexnames` By default the targets have names built with `\thepage`: `page.\thepage`, so e.g., `page.4` or `page.iii`. The names require that every page as an unique number representation. A frequent problem here is with title pages which often don't show page numbers but internally use the same number as a following page. If you get messages about `destination with the same identifier (name`page.1`)`, change the number representation of title pages or disable the page target.

If the boolean option `hypertexnames` is set to false, an internal page counter is stepped and used as arabic number. In most cases this should mean that you get the absolute page number (exceptions could be with documents throwing away or duplicating pages at shipout). This option avoids the problem with duplicated identifiers, but does not work with an index, as the backlinks have to be created from the represention passed to the index.

plainpages This forces page anchors to be named by the arabic form, but unlike the previous option it does not use an internal counter but the page counter. That means if you have for example roman and arabic pages you will get duplicated target names as page i and page 1 both set the anchor page.1.

`\@currentHpage`

Starting with version 7.01c hyperref stores the name of the page target it has just set into the (global) variable `\@currentHpage`. The fallback value is `Doc-Start`. As the target is set when the page is shipout, `\@currentHpage` is only reliable at shipout. With a LaTeX format from 2023-11-01 the name can be labeled like this:

```
%labeling in the document
\RecordProperties{mylabel}{pagetarget}
%linking to the page:
\hyperlink{\RefProperty{mylabel}{pagetarget}}{some text}
```

5.4 Configuration options

raiselinks	boolean	*true*	In the hypertex driver, the height of links is normally calculated by the driver as simply the base line of contained text; this options forces `\special` commands to reflect the real height of the link (which could contain a graphic)
breaklinks	boolean	*both*	This option is in hyperref only used in the dviwindo driver, in all other cases it doesn't do anything sensible—it neither allows nor prevents links to be broken. The ocgx2 package checks the state of the boolean.
pageanchor	boolean	*true*	Determines whether every page is given an implicit anchor at the top left corner. If this is turned off, `\printindex` will not contain valid hyperlinks.
nesting	boolean	*false*	Allows links to be nested; no drivers currently support this.

Note for option breaklinks: The correct value is automatically set according to the driver features. It can be overwritten for drivers that do not support broken links. However, at any case, the link area will be wrong and displaced.

5.5 Backend drivers

If no driver is specified, the package tries to find a driver in the following order:

1. Autodetection, some TeX processors can be detected at TeX macro level (pdfTeX, XeTeX, VTeX).
2. Option driverfallback. If this option is set, its value is taken as driver option.
3. Macro `\Hy@defaultdriver`. The macro takes a driver file name (without file extension).
4. Package default is hypertex.

Many distributions are using a driver file hypertex.cfg that define `\Hy@defaultdriver` with hdvips. This is recommended because driver dvips provides much more features than hypertex for PDF generation.

driverfallback	Its value is used as driver option if the driver is not given or autodetected.
dvipdfm	Sets up hyperref for use with the dvipdfm driver.
dvipdfmx	Sets up hyperref for use with the dvipdfmx driver.
dvips	Sets up hyperref for use with the dvips driver.
dvipsone	Sets up hyperref for use with the dvipsone driver.
dviwindo	Sets up hyperref for use with the dviwindo Windows previewer.
hypertex	Sets up hyperref for use with the HyperTEX-compliant drivers.
latex2html	Redefines a few macros for compatibility with latex2html.
nativepdf	An alias for dvips
pdfmark	An alias for dvips
pdftex	Sets up hyperref for use with the pdftex program.
ps2pdf	Redefines a few macros for compatibility with Ghostscript's PDF writer, otherwise identical to dvips.
tex4ht	For use with TEX4ht
textures	For use with Textures
vtex	For use with MicroPress' VTeX; the PDF and HTML backends are detected automatically.
vtexpdfmark	For use with VTeX's PostScript backend.
xetex	For use with XeTEX (using backend for dvipdfm).

If you use dviwindo, you may need to redefine the macro \wwwbrowser (the default is C:\netscape\netscape) to tell dviwindo what program to launch. Thus, users of Internet Explorer might add something like this to hyperref.cfg:

```
\renewcommand{\wwwbrowser}{C:\string\Program\space
 Files\string\Plus!\string\Microsoft\space
 Internet\string\iexplore.exe}
```

5.6 Extension options

extension	text		Set the file extension (e.g. dvi) which will be appended to file links created if you use the xr package.
hyperfigures	boolean		
backref	text	*false*	Adds 'backlink' text to the end of each item in the bibliography, as a list of section numbers. This can only work properly *if* there is a blank line after each \bibitem. Supported values are section, slide, page, none, or false. If no value is given, section is taken as default.
pagebackref	boolean	*false*	Adds 'backlink' text to the end of each item in the bibliography, as a list of page numbers.
hyperindex	boolean	*true*	Makes the page numbers of index entries into hyperlinks. Relays on unique page anchors (pageanchor, ...) pageanchors and plainpages=false.
hyperfootnotes	boolean	*true*	Makes the footnote marks into hyperlinks to the footnote text. Easily broken ...
encap			Sets encap character for hyperindex
linktoc	text	*section*	make text (section), page number (page), both (all) or nothing (none) be link on TOC, LOF and LOT

linktocpage	boolean	*false*	make page number, not text, be link on TOC, LOF and LOT
breaklinks	boolean	*false*	allow links to break over lines by making links over multiple lines into PDF links to the same target
colorlinks	boolean	*false*	Colors the text of links and anchors. The colors chosen depend on the the type of link. At present the only types of link distinguished are citations, page references, URLs, local file references, and other links. Unlike colored boxes, the colored text remains when printing.
linkcolor	color	*red*	Color for normal internal links.
anchorcolor	color	*black*	Color for anchor text. Ignored by most drivers.
citecolor	color	*green*	Color for bibliographical citations in text.
filecolor	color	*cyan*	Color for URLs which open local files.
menucolor	color	*red*	Color for Acrobat menu items.
runcolor	color	*filecolor*	Color for run links (launch annotations).
urlcolor	color	*magenta*	Color for linked URLs.
allcolors	color		Set all color options (without border and field options).
frenchlinks	boolean	*false*	Use small caps instead of color for links.
hidelinks			Hide links (removing color and border).

Note that all color names must be defined before use, following the normal system of the standard LaTeX color package.

5.7 PDF-specific display options

bookmarks	boolean	*true*	A set of Acrobat bookmarks are written, in a manner similar to the table of contents, requiring two passes of LaTeX. Some postprocessing of the bookmark file (file extension .out) may be needed to translate LaTeX codes, since bookmarks must be written in PDFEncoding. To aid this process, the .out file is not rewritten by LaTeX if it is edited to contain a line `\let\WriteBookmarks\relax`
bookmarksopen	boolean	*false*	If Acrobat bookmarks are requested, show them with all the subtrees expanded.
bookmarksopenlevel	parameter		level (`\maxdimen`) to which bookmarks are open
bookmarksnumbered	boolean	*false*	If Acrobat bookmarks are requested, include section numbers.
bookmarkstype	text	*toc*	to specify which 'toc' file to mimic

CJKbookmarks	boolean	*false*	This option should be used to produce CJK bookmarks. Package hyperref supports both normal and preprocessed mode of the CJK package; during the creation of bookmarks, it simply replaces CJK's macros with special versions which expand to the corresponding character codes. Note that without the 'unicode' option of hyperref you get PDF files which actually violate the PDF specification because non-Unicode character codes are used – some PDF readers localized for CJK languages (most notably Acroread itself) support this. Also note that option 'CJKbookmarks' cannot be used together with option 'unicode'. No mechanism is provided to translate non-Unicode bookmarks to Unicode; for portable PDF documents only Unicode encoding should be used.
pdfhighlight	name	*/I*	How link buttons behave when selected; /I is for inverse (the default); the other possibilities are /N (no effect), /O (outline), and /P (inset highlighting).
citebordercolor	RGB color	*0 1 0*	The color of the box around citations
filebordercolor	RGB color	*0 .5 .5*	The color of the box around links to files
linkbordercolor	RGB color	*1 0 0*	The color of the box around normal links
menubordercolor	RGB color	*1 0 0*	The color of the box around Acrobat menu links
urlbordercolor	RGB color	*0 1 1*	The color of the box around links to URLs
runbordercolor	RGB color	*0 .7 .7*	Color of border around 'run' links
allbordercolors			Set all border color options
pdfborder		*0 0 1*	The style of box around links; defaults to a box with lines of 1pt thickness, but the colorlinks option resets it to produce no border.

The color of link borders used to be specified *only* as 3 numbers in the range 0..1, giving an RGB color. Since version 6.76a, the usual color specifications of package (x)color can be used if xcolor has been loaded. For further information see description of package hycolor.

The bookmark commands are stored in a file called *jobname*.out. The files is not processed by LaTeX so any markup is passed through. You can postprocess this file as needed; as an aid for this, the .out file is not overwritten on the next TeX run if it is edited to contain the line

```
\let\WriteBookmarks\relax
```

5.8 PDF display and information options

baseurl	URL		Sets the base URL of the PDF document
pdfpagemode	name	*empty*	Determines how the file is opening in Acrobat; the possibilities are UseNone, UseThumbs (show thumbnails), UseOutlines (show bookmarks), FullScreen, UseOC (PDF 1.5), and UseAttachments (PDF 1.6). If no mode if explicitly chosen, but the bookmarks option is set, UseOutlines is used.
pdftitle	text		Sets the document information Title field

pdfauthor	text		Sets the document information Author field
pdfsubject	text		Sets the document information Subject field
pdfcreator	text		Sets the document information Creator field
pdfcreationdate	date		Sets the creation date, see below the table for more info about the format
pdfmoddate	date		Sets the modification date, see below the table for more info about the format
addtopdfcreator	text		Adds additional text to the document information Creator field
pdfkeywords	text		Sets the document information Keywords field
pdftrapped	name	*empty*	Sets the document information Trapped entry. Possible values are `True`, `False` and `Unknown`. An empty value means, the entry is not set.
pdfinfo	key value list	*empty*	Alternative interface for setting the document information.
pdfview	name	*XYZ*	Sets the default PDF 'view' for each link
pdfstartpage	integer	*1*	Determines on which page the PDF file is opened. An empty value means, the entry is not set.
pdfstartview	name	*Fit*	Set the startup page view
pdfremotestartview	name	*Fit*	Set the startup page view of remote PDF files
pdfpagescrop	n n n n		Sets the default PDF crop box for pages. This should be a set of four numbers
pdfcenterwindow	boolean	*false*	position the document window in the center of the screen
pdfdirection	name	*empty*	direction setting. Possible values: `L2R` (left to right) and `R2L` (right to left)
pdfdisplaydoctitle	boolean	*false*	display document title instead of file name in title bar
pdfduplex	name	*empty*	paper handling option for print dialog. Possible vatues are: `Simplex` (print single-sided), `DuplexFlipShortEdge` (duplex and flip on the short edge of the sheet), `DuplexFlipLongEdge` (duplex and flip on the long edge of the sheet)
pdffitwindow	boolean	*false*	resize document window to fit document size
pdflang	name	*relax*	PDF language identifier (RFC 3066)
pdfmenubar	boolean	*true*	make PDF viewer's menu bar visible
pdfnewwindow	boolean	*false*	make links that open another PDF file start a new window
pdfnonfullscreenpagemode	name	*empty*	page mode setting on exiting full-screen mode. Possible values are `UseNone`, `UseOutlines`, `UseThumbs`, and `UseOC`
pdfnumcopies	integer	*empty*	number of printed copies
pdfpagelayout	name	*empty*	set layout of PDF pages. Possible values: `SinglePage`, `OneColumn`, `TwoColumnLeft`, `TwoColumnRight`, `TwoPageLeft`, and `TwoPageRight`
pdfpagelabels	boolean	*true*	set PDF page labels

pdfpagetransition	name	*empty*	set PDF page transition style. Possible values are Split, Blinds, Box, Wipe, Dissolve, Glitter, R, Fly, Push, Cover, Uncover, Fade. The default according to the PDF Reference is R, which simply replaces the old page with the new one.
pdfpicktraybypdfsize	boolean	*false*	specify whether PDF page size is used to select input paper tray in print dialog
pdfprintarea	name	*empty*	set /PrintArea of viewer preferences. Possible values are MediaBox, CropBox, BleedBox, TrimBox, and ArtBox. The dafault according to the PDF Refence is CropBox
pdfprintclip	name	*empty*	set /PrintClip of viewer preferences. Possible values are MediaBox, CropBox, BleedBox, TrimBox, and ArtBox. The dafault according to the PDF Refence is CropBox
pdfprintpagerange	n n (n n)*	*empty*	set /PrintPageRange of viewer preferences
pdfprintscaling	name	*empty*	page scaling option for print dialog (option /PrintScaling of viewer preferences, PDF 1.6); valid values are None and AppDefault
pdftoolbar	boolean	*true*	make PDF toolbar visible
pdfviewarea	name	*empty*	set /ViewArea of viewer preferences. Possible values are MediaBox, CropBox, BleedBox, TrimBox, and ArtBox. The dafault according to the PDF Refence is CropBox
pdfviewclip	name	*empty*	set /ViewClip of viewer preferences Possible values are MediaBox, CropBox, BleedBox, TrimBox, and ArtBox. The dafault according to the PDF Refence is CropBox
pdfwindowui	boolean	*true*	make PDF user interface elements visible
unicode	boolean	*true*	Unicode encoded PDF strings

The dates CreationDate and ModDate are normally set automatically to the current date/time when the compilation started. If they should be changed (e.g. for regression tests to produce reproducible documents) they can be set with \hypersetup with the keys pdfcreationdate, pdfmoddate or externally by setting the SOURCE_DATE_EPOCH environment variable.

```
\hypersetup{pdfcreationdate=D:20010101205959-00'00'}
```

The format should be a full date/time in PDF format, so one of these (naturally the numbers can change):

```
D:20010101205959-00'00'
D:20010101205959+00'00'
D:20010101205959Z
```

Each link in Acrobat carries its own magnification level, which is set using PDF coordinate space, which is not the same as TEX's. The unit is bp and the origin is in the lower left corner. See also \hypercalcbp that is explained on page 27. pdfTEX works by supplying default values for XYZ (horizontal × vertical × zoom) and FitBH. However, drivers using pdfmark do not supply

defaults, so hyperref passes in a value of -32768, which causes Acrobat to set (usually) sensible defaults. The following are possible values for the pdfview, pdfstartview and pdfremotestartview parameters.

XYZ	*left top zoom*	Sets a coordinate and a zoom factor. If any one is null, the source link value is used. *null null null* will give the same values as the current page.
Fit		Fits the page to the window.
FitH	*top*	Fits the width of the page to the window.
FitV	*left*	Fits the height of the page to the window.
FitR	*left bottom right top*	Fits the rectangle specified by the four coordinates to the window.
FitB		Fits the page bounding box to the window.
FitBH	*top*	Fits the width of the page bounding box to the window.
FitBV	*left*	Fits the height of the page bounding box to the window.

The pdfpagelayout can be one of the following values.

SinglePage	Displays a single page; advancing flips the page
OneColumn	Displays the document in one column; continuous scrolling.
TwoColumnLeft	Displays the document in two columns, odd-numbered pages to the left.
TwoColumnRight	Displays the document in two columns, odd-numbered pages to the right.
TwoPageLeft	Displays two pages, odd-numbered pages to the left (since PDF 1.5).
TwoPageRight	Displays two pages, odd-numbered pages to the right (since PDF 1.5).

Finally, the pdfpagetransition can be one of the following values, where */Di* stands for direction of motion in degrees, generally in 90 degree steps, */Dm* is a horizontal (/H) or vertical (/V) dimension (e.g. Blinds /Dm /V), and */M* is for motion, either in (/I) or out (/O).

Blinds	/Dm	Multiple lines distributed evenly across the screen sweep in the same direction to reveal the new page.
Box	/M	A box sweeps in or out.
Dissolve		The page image dissolves in a piecemeal fashion to reveal the new page.
Glitter	/Di	Similar to Dissolve, except the effect sweeps across the screen.
Split	/Dm /M	Two lines sweep across the screen to reveal the new page.
Wipe	/Di	A single line sweeps across the screen to reveal the new page.
R		Simply replaces the old page with the new one.
Fly	/Di /M	Changes are flown out or in (as specified by /M), in the direction specified by /Di, to or from a location that is offscreen except when /Di is None.

Push	/Di	The old page slides off the screen while the new page slides in, pushing the old page out in the direction specified by /Di.
Cover	/Di	The new page slides on to the screen in the direction specified by /Di, covering the old page.
Uncover	/Di	The old page slides off the screen in the direction specified by /Di, uncovering the new page in the direction specified by /Di.
Fade		The new page gradually becomes visible through the old one.

5.9 Option pdfinfo

The information entries can be set using pdftitle, pdfsubject, ….. Option pdfinfo provides an alternative interface. It takes a key value list. The key names are the names that appear in the PDF information dictionary directly. Known keys such as Title, Subject, Trapped and other are mapped to options pdftitle, subject, trapped, …Unknown keys are added to the information dictionary. Their values are text strings (see PDF specification). Example:

```
\hypersetup{
 pdfinfo={
  Title={My Title},
  Subject={My Subject},
  NewKey={Foobar},
  % ...
 }
}
```

5.10 Big alphabetical list

The following is a complete listing of available options for hyperref, arranged alphabetically.

allbordercolors		Set all border color options
allcolors		Set all color options (without border and field options)
anchorcolor	*black*	set color of anchors, ignored by most drivers.
backref	*false*	do bibliographical back references
baseurl	*empty*	set base URL for document
bookmarks	*true*	make bookmarks
bookmarksnumbered	*false*	put section numbers in bookmarks
bookmarksopen	*false*	open up bookmark tree
bookmarksopenlevel	\maxdimen	level to which bookmarks are open
bookmarkstype	*toc*	to specify which 'toc' file to mimic
breaklinks	*false*	allow links to break over lines
CJKbookmarks	*false*	to produce CJK bookmarks
citebordercolor	*0 1 0*	color of border around cites
citecolor	*green*	color of citation links
colorlinks	*false*	color links
	true	(tex4ht, dviwindo)
debug	*false*	provide details of anchors defined; same as verbose

destlabel	*false*	destinations are named by the first \label after the anchor creation
draft	*false*	do not do any hyperlinking
driverfallback		default if no driver specified or detected
dvipdfm		use dvipdfm backend
dvipdfmx		use dvipdfmx backend
dvips		use dvips backend
dvipsone		use dvipsone backend
dviwindo		use dviwindo backend
encap		to set encap character for hyperindex
extension	*dvi*	suffix of linked files
filebordercolor	*0 .5 .5*	color of border around file links
filecolor	*cyan*	color of file links
final	*true*	opposite of option draft
frenchlinks	*false*	use small caps instead of color for links
hidelinks		Hide links (removing color and border)
hyperfigures	*false*	make figures hyper links
hyperfootnotes	*true*	set up hyperlinked footnotes
hyperindex	*true*	set up hyperlinked indices
hypertex		use HyperTeX backend
hypertexnames	*true*	use guessable names for links
implicit	*true*	redefine LaTeX internals
latex2html		use LaTeX2HTML backend
linkbordercolor	*1 0 0*	color of border around links
linkcolor	*red*	color of links
linktoc	*section*	make text be link on TOC, LOF and LOT
linktocpage	*false*	make page number, not text, be link on TOC, LOF and LOT
menubordercolor	*1 0 0*	color of border around menu links
menucolor	*red*	color for menu links
nativepdf	*false*	an alias for dvips
naturalnames	*false*	use LaTeX-computed names for links
nesting	*false*	allow nesting of links
next-anchor		allow to set the name of the next anchor
pageanchor	*true*	put an anchor on every page
pagebackref	*false*	backreference by page number
pdfauthor	*empty*	text for PDF Author field
pdfborder	*0 0 1*	width of PDF link border
	0 0 0	(colorlinks)
pdfborderstyle		border style for links
pdfcenterwindow	*false*	position the document window in the center of the screen
pdfcreator	*LaTeX with hyperref*	text for PDF Creator field
pdfcreationdate		Sets the creation date, see below the table in section 5.8 for more info about the format
pdfdirection	*empty*	direction setting
pdfdisplaydoctitle	*false*	display document title instead of file name in title bar
pdfduplex	*empty*	paper handling option for print dialog
pdffitwindow	*false*	resize document window to fit document size
pdfhighlight	*/I*	set highlighting of PDF links

pdfinfo	*empty*	alternative interface for setting document information
pdfkeywords	*empty*	text for PDF Keywords field
pdflang	*relax*	PDF language identifier (RFC 3066)
pdfmark	*false*	an alias for dvips
pdfmenubar	*true*	make PDF viewer's menu bar visible
pdfmoddate		Sets the modification date, see below the table in section 5.8 for more info about the format
pdfnewwindow	*false*	make links that open another PDF file start a new window
pdfnonfullscreenpagemode	*empty*	page mode setting on exiting full-screen mode
pdfnumcopies	*empty*	number of printed copies
pdfpagelabels	*true*	set PDF page labels
pdfpagelayout	*empty*	set layout of PDF pages
pdfpagemode	*empty*	set default mode of PDF display
pdfpagescrop	*empty*	set crop size of PDF document
pdfpagetransition	*empty*	set PDF page transition style
pdfpicktraybypdfsize	*empty*	set option for print dialog
pdfprintarea	*empty*	set /PrintArea of viewer preferences
pdfprintclip	*empty*	set /PrintClip of viewer preferences
pdfprintpagerange	*empty*	set /PrintPageRange of viewer preferences
pdfprintscaling	*empty*	page scaling option for print dialog
pdfproducer	*empty*	text for PDF Producer field
pdfremotestartview	*Fit*	starting view of remote PDF documents
pdfstartpage	*1*	page at which PDF document opens
pdfstartview	*Fit*	starting view of PDF document
pdfsubject	*empty*	text for PDF Subject field
pdftex		use pdfTeX backend
pdftitle	*empty*	text for PDF Title field
pdftoolbar	*true*	make PDF toolbar visible
pdftrapped	*empty*	Sets the document information Trapped entry. Possible values are `True`, `False` and `Unknown`. An empty value means, the entry is not set.
pdfview	*XYZ*	PDF 'view' when on link traversal
pdfviewarea	*empty*	set /ViewArea of viewer preferences
pdfviewclip	*empty*	set /ViewClip of viewer preferences
pdfwindowui	*true*	make PDF user interface elements visible
plainpages	*false*	do page number anchors as plain Arabic
ps2pdf		use ps2pdf backend
psdextra	*false*	define more short names for PDF string commands
raiselinks	*false*	raise up links (for HyperTeX backend)
runbordercolor	*0 .7 .7*	color of border around 'run' links
runcolor	*filecolor*	color of 'run' links
setpagesize	*true*	set page size by special driver commands
tex4ht		use TeX4ht backend
textures		use Textures backend
unicode	*true*	Unicode encoded pdf strings, starting with version v7.00g set by default to true for all engines. It will load a number of definitions in puenc.def. It can be set to false for pdflatex, but this is not recommended.

urlbordercolor	*0 1 1*	color of border around URL links
urlcolor	*magenta*	color of URL links
verbose	*false*	be chatty
vtex		use VTeX backend
xetex		use XeTeX backend

6 Additional user macros

If you need to make references to URLs, or write explicit links, the following low-level user macros are provided:

`\href[`*options*`]{`*URL*`}{`*text*`}`

The *text* is made a hyperlink to the *URL*; this must be a full URL (relative to the base URL, if that is defined). The special characters # and % do *not* need to be escaped in any way (unless the command is used in the argument of another command).

The optional argument *options* recognizes the hyperref options `pdfremotestartview`, `pdfnewwindow` and the following key value options:

page: Specifies the start page number of remote PDF documents. First page is `1`.

ismap: Boolean key, if set to `true`, the URL should appended by the coordinates as query parameters by the PDF viewer.

nextactionraw: The value of key `/Next` of action dictionaries, see PDF specification.

`\url{`*URL*`}`

Similar to `\href{`*URL*`}{\nolinkurl{`*URL*`}}`. Depending on the driver `\href` also tries to detect the link type. Thus the result can be a url link, file link, ….. The implementation makes use of the url package and its commands can be used to format and fine tune the url, but the behaviour of `\url` is different to the behaviour in the url package: Most importantly the hyperref `\url` expands commands. If the behaviour or the original `\url` is needed, the url package should be loaded first and the command should be copied into some alias command.

`\nolinkurl{`*URL*`}`

Write *URL* in the same way as `\url` described above, without creating a hyperlink.

`\hyperbaseurl{`*URL*`}`

A base *URL* is established, which is prepended to other specified URLs, to make it easier to write portable documents.

`\hyperimage{`*imageURL*`}{`*text*`}`

The link to the image referenced by the URL is inserted, using *text* as the anchor.

For drivers that produce HTML, the image itself is inserted by the browser, with the *text* being ignored completely.

`\hyperdef{`*category*`}{`*name*`}{`*text*`}`

A target area of the document (the *text*) is marked, and given the name *category.name*

`\hyperref{`*URL*`}{`*category*`}{`*name*`}{`*text*`}`

text is made into a link to *URL*#*category.name*

`\hyperref[`*label*`]{`*text*`}`

text is made into a link to the same place as `\ref{`*label*`}` would be linked.

`\hyperlink{`*name*`}{`*text*`}`

`\hypertarget{`*name*`}{`*text*`}`

A simple internal link is created with `\hypertarget`, with two parameters of an anchor *name*, and anchor *text*. `\hyperlink` has two arguments, the name of a hypertext object defined somewhere by `\hypertarget`, and the *text* which be used as the link on the page.

Note that in HTML parlance, the `\hyperlink` command inserts a notional # in front of each link, making it relative to the current testdocument; `\href` expects a full URL.

`\phantomsection`

This sets an anchor at this location. It works similar to `\hypertarget{}{}` with an automatically chosen anchor name. Often it is used in conjunction with `\addcontentsline` for sectionlike things (index, bibliography, preface). `\addcontentsline` refers to the latest previous location where an anchor is set. Example:

```
\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.

`\hyperget{`*anchor*`}{`*label*`}` `\hyperget{`*pageanchor*`}{`*label*`}`

This retrieves the anchor or the page anchor from a label in an expandable way. It takes `\HyperDestNameFilter` into account. It can e.g. be used with the `\bookmark` from the bookmark package to set a destination to a label:

```
\bookmark[dest=\hyperget{anchor}{sec}]{section}
```

As *pageanchor* retrieves the page number from the label it can't be use together with the option `plainpages`.

`\hyperget{`*currentanchor*`}{}`

This retrieves the last anchor that has been set. It too takes `\HyperDestNameFilter` into account.

\autoref{*label*}

This is a replacement for the usual \ref command that places a contextual label in front of the reference. This gives your users a bigger target to click for hyperlinks (e.g. 'section 2' instead of merely the number '2').

The label is worked out from the context of the original \label command by hyperref by using the macros listed below (shown with their default values). The macros can be (re)defined in documents using \(re)newcommand; note that some of these macros are already defined in the standard document classes. The mixture of lowercase and uppercase initial letters is deliberate and corresponds to the author's practice.

For each macro below, hyperref checks *autorefname before *name. For instance, it looks for \figureautorefname before \figurename.

Macro	*Default*
\figurename	Figure
\tablename	Table
\partname	Part
\appendixname	Appendix
\equationname	Equation
\Itemname	item
\chaptername	chapter
\sectionname	section
\subsectionname	subsection
\subsubsectionname	subsubsection
\paragraphname	paragraph
\Hfootnotename	footnote
\AMSname	Equation
\theoremname	Theorem
\page	page

Example for a redefinition if babel is used:

```
\usepackage[ngerman]{babel}
\addto\extrasngerman{%
  \def\subsectionautorefname{Unterkapitel}%
}
```

Hint: \autoref works via the counter name that the reference is based on. Sometimes \autoref chooses the wrong name, if the counter is used for different things. For example, it happens with \newtheorem if a lemma shares a counter with theorems. Then package aliascnt provides a method to generate a simulated second counter that allows the differentiation between theorems and lemmas:

```
\documentclass{article}

\usepackage{aliascnt}
\usepackage{hyperref}

\newtheorem{theorem}{Theorem}

\newaliascnt{lemma}{theorem}
\newtheorem{lemma}[lemma]{Lemma}
\aliascntresetthe{lemma}
```

```
\providecommand*{\lemmaautorefname}{Lemma}

\begin{document}

We will use \autoref{a} to prove \autoref{b}.

\begin{lemma}\label{a}
  Nobody knows.
\end{lemma}

\begin{theorem}\label{b}
  Nobody is right.
\end{theorem}.

\end{document}
```

\autopageref{*label*}

It replaces \pageref and adds the name for page in front of the page reference. First \pageautorefname is checked before \pagename.

For instances where you want a reference to use the correct counter, but not to create a link, there are starred forms (these starred forms exist even if hyperref has been loaded with implicit=false):

\ref*{*label*}

\pageref*{*label*}

\autoref*{*label*}

\autopageref*{*label*}

A typical use would be to write

```
\hyperref[other]{that nice section (\ref*{other}) we read before}
```

We want \ref*{other} to generate the correct number, but not to form a link, since we do this ourselves with \hyperref.

\pdfstringdef{*macroname*}{*TEXstring*}

\pdfstringdef returns a macro containing the PDF string. (Currently this is done globally, but do not rely on it.) All the following tasks, definitions and redefinitions are made in a group to keep them local:

- Switching to PD1 or PU encoding
- Defining the "octal sequence commands" (\345): \edef\3{\string\3}

- Special glyphs of TEX: \{, \%, \&, \space, \dots, etc.
- National glyphs (german.sty, french.sty, etc.)
- Logos: \TeX, \eTeX, \MF, etc.
- Disabling commands that do not provide useful functionality in bookmarks: \label, \index, \glossary, \discretionary, \def, \let, etc.
- LATEX's font commands like \textbf, etc.
- Support for \xspace provided by the xspace package

In addition, parentheses are protected to avoid the danger of unsafe unbalanced parentheses in the PDF string. For further details, see Heiko Oberdiek's EuroTEX paper distributed with hyperref.

\begin{NoHyper}…\end{NoHyper}

Sometimes we just don't want the wretched package interfering with us. Define an environment we can put in manually, or include in a style file, which stops the hypertext functions doing anything. This is used, for instance, in the Elsevier classes, to stop hyperref playing havoc in the front matter.

6.1 Bookmark macros

6.1.1 Setting bookmarks

Usually hyperref automatically adds bookmarks for \section and similar macros. But they can also set manually.

\pdfbookmark[*level*]{text}{*name*}

creates a bookmark with the specified text and at the given level (default is 0). As name for the internal anchor name is used (in conjunction with level). Therefore the name must be unique (similar to \label).

\currentpdfbookmark{*text*}{*name*}

creates a bookmark at the current level.

\subpdfbookmark{*text*}{*name*}

creates a bookmark one step down in the bookmark hierarchy. Internally the current level is increased by one.

\belowpdfbookmark{*text*}{*name*}

creates a bookmark below the current bookmark level. However after the command the current bookmark level has not changed.
Hint: Package bookmark replaces hyperref's bookmark organization by a new algorithm:

- Usually only one LATEX run is needed.
- More control over the bookmark appearance (color, font).

- Different bookmark actions are supported (external file links, URLs, …).

Therefore I recommend using this package.

6.1.2 Replacement macros

hyperref takes the text for bookmarks from the arguments of commands like \section, which can contain things like math, colors, or font changes, none of which will display in bookmarks as is.

\texorpdfstring{*TEXstring*}{*PDFstring*}

For example,

```
\section{Pythagoras:
  \texorpdfstring{$ a^2 + b^2 = c^2 $}{%
    a\texttwosuperior\ + b\texttwosuperior\ =
    c\texttwosuperior
  }%
}
\section{\texorpdfstring{\textcolor{red}}{}{Red} Mars}
```

\pdfstringdef executes the hook before it expands the string. Therefore, you can use this hook to perform additional tasks or to disable additional commands.

```
\expandafter\def\expandafter\pdfstringdefPreHook
\expandafter{%
  \pdfstringdefPreHook
  \renewcommand{\mycommand}[1]{}%
}
```

However, for disabling commands, an easier way is via \pdfstringdefDisableCommands, which adds its argument to the definition of \pdfstringdefPreHook ('@' can here be used as letter in command names):

```
\pdfstringdefDisableCommands{%
  \let~\textasciitilde
  \def\url{\pdfstringdefWarn\url}%
  \let\textcolor\@gobble
}
```

6.2 Pagelabels

\thispdfpagelabel{*page number format*}

This allows to change format of the page number shown in the tool bar of a PDF viewer for a specific page, for example

\thispdfpagelabel{Empty Page-\roman{page}}

The command affects the page on which it is executed, so asynchronous page breaking should be taken into account. It should be used in places where for example \thispagestyle can be use too.

6.3 Utility macros

`\hypercalcbp{`*dimen specification*`}`

`\hypercalcbp` takes a TEX dimen specification and converts it to bp and returns the number without the unit. This is useful for options `pdfview`, `pdfstartview` and `pdfremotestartview`. Example:

```
\hypersetup{
 pdfstartview={FitBH \hypercalcbp{\paperheight-\topmargin-1in
  -\headheight-\headsep}
}
```

The origin of the PDF coordinate system is the lower left corner.

Note, for calculations you need either package calc or ε-TEX. Nowadays the latter should automatically be enabled for LATEX formats. Users without ε-TEX, please, look in the source documentation `hyperref.dtx` for further limitations.

Also `\hypercalcbp` cannot be used in option specifications of `\documentclass` and `\usepackage`, because LATEX expands the option lists of these commands. However package hyperref is not yet loaded and an undefined control sequence error would arise.

7 New Features[5]

7.1 Option 'pdflinkmargin'

Option 'pdflinkmargin' is an experimental option for specifying a link margin, if the driver supports this. Default is 1 pt for supporting drivers.

pdfTeX
- The link area also depends on the surrounding box.
- Settings have local effect.
- When a page is shipped out, pdfTeX uses the current setting of the link margin for all links on the page.

pdfmark
- Settings have global effect.

xetex
- Settings must be done in the preamble or the first page and then have global effect. The key inserts the new (x)dvipdfmx special `\special{dvipdfmx:config g #1}` (with the unit removed).

Other drivers Unsupported.

7.2 Field option 'calculatesortkey'

Fields with calculated values are calculated in document order by default. If calculated field values depend on other calculated fields that appear later in the document, then the correct calculation order can be specified with option 'calculatesortkey'. Its value is used as key to lexicographically sort the calculated fields. The sort key do not need to be unique. Fields that share the same key are sorted in document order.

Currently the field option 'calculatesortkey' is only supported by the driver for pdfTeX.

[5] This section moved from the README file, needs more integration into the manual

7.3 Option 'next-anchor'

This option allows to overwrite the anchor name of the next anchor. This makes it possible to give for example the heading of the table of contents an anchor name which can be referenced with a `\bookmark` command from the bookmark package:

```
\hypersetup{next-anchor=toc}
\tableofcontents
\bookmark[dest=\HyperDestNameFilter{toc},level=section]{\contentsname}
```

7.4 Option 'localanchorname'

When an anchor is set (e.g. via `\refstepcounter`, then the anchor name is globally set to the current anchor name.

Deprecated 2022-04-27 v7.00o

For example:

```
\section{Foobar}
\begin{equation}\end{equation}
\label{sec:foobar}
```

With the default global setting (localanchorname=false) a reference to 'sec:foobar' jumps to the equation before. With option 'localanchorname' the anchor of the equation is forgotten after the environment and the reference 'sec:foobar' jumps to the section title.

Option 'localanchorname' is an experimental option, there might be situations, where the anchor name is not available as expected.

The option is deprecated: it makes it difficult for package authors to add targets for links if it is unclear if `\@currentHref` is set locally or globally.

7.5 Option 'customdriver'

The value of option 'customdriver' is the name of an external driver file without extension '.def'. The file must have `\ProvidesFile` with a version date and number that match the date and number of 'hyperref', otherwise a warning is given.

Because the interface, what needs to be defined in the driver, is not well defined and quite messy, the option is mainly intended to ease developing, testing, debugging the driver part.

7.6 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`.

7.7 \XeTeXLinkBox

When XeTeX generates a link annotation, it does not look at the boxes (as the other drivers), but only at the character glyphs. If there are no glyphs (images, rules, ...), then it does not generate a link annotation. Macro `\XeTeXLinkBox` puts its argument in a box and adds spaces at the lower left and upper right corners. An additional margin can be specified by setting it to the dimen register `\XeTeXLinkMargin`. The default is 2pt.

Example:

```
% xelatex
\documentclass{article}
\usepackage{hyperref}
\setlength{\XeTeXLinkMargin}{1pt}
\begin{document}
\section{Hello World}
\newpage
\label{sec:hello}
\hyperref[sec:hello]{%
  \XeTeXLinkBox{\rule{10mm}{10mm}}%
}
\end{document}
```

7.8 \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 'stoppedearly' is available. It is enabled if \MaybeStopEarly or \MaybeStopNow end hyperref prematurely.

7.9 \unichar

If a Unicode character is not supported by puenc.def, it can be given by using \unichar. Its 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)
```

7.10 \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 Hàn Thế Thành. 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%
  }%
 }%
}
```

7.11 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`.)

7.12 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.

A better implementation which hasn't the disadvantage to prevent line breaks is in the ocgx2 package. Check its documentation for details how to use it.

- 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 of 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, LuaTeX and dvipdfmx.

7.13 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 ([PDF/A 6.6.1].
- Restricted: Named actions (NextPage, PrevPage, FirstPage, LastPage) [PDF/A 6.6.1].
- Many things are disabled in PDF formulars:
  - 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}`).

7.14 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)

7.15 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.

7.16 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)

7.17 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.

7.18 Field option 'name'

Many form objects uses the label argument for several purposes:

- Layouted 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.

7.19 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 (outside ascii if an unicode engine is used) and Unicode otherwise. This option is not intended for the unicode engines.

All drivers use unicode by default now. If another encoding should be forced, it should be done in `hypersetup`.

7.20 Color options/package hycolor

See documentation of package 'hycolor'.

7.21 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).

7.22 Starred form of \autoref

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

7.23 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

7.24 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
```

7.25 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, therefore 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.

7.26 Default driver setting

(hyperref $\geq$ 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'.

7.27 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}
```

7.28 \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.

7.29 puenc encoding, puenc-greekbasic.def and puenc-extra.def

The `unicode` option loads for the bookmarks `puenc.def` which contains quite a lot definitions of commands for the bookmarks. As `unicode` is now true for all engines, this file is now also loaded with pdflatex. Some of the definitions in `puenc.def` clash with other uses. To reduce the impact hyperref uses two strategies.

- A number of command are only defined conditionally: The commands for the cyrillic block if `\CYRDZE` is defined, greek if `\textBeta` is defined, and hebrew if `\hebdalet` is defined.

  The greek block is in an extra file, `puenc-greekbasic.def`, which can be loaded manually if needed.

- Other commands are moved to an extra file `puenc-extra.def` which is not loaded automatically, but can be loaded in the preamble if needed. Currently this file contains all definitions for the accent `\G`.

8 Acrobat-specific behavior

If you want to access the menu options of Acrobat Reader or Exchange, the following macro is provided in the appropriate drivers:

`\Acrobatmenu{`*menuoption*`}{`*text*`}`

The *text* is used to create a button which activates the appropriate *menuoption*. The following table lists the option names you can use—comparison of this with the menus in Acrobat Reader or Exchange will show what they do. Obviously some are only appropriate to Exchange.

File	Open, Close, Scan, Save, SaveAs, Optimizer:SaveAsOpt, Print, PageSetup, Quit
File→Import	ImportImage, ImportNotes, AcroForm:ImportFDF
File→Export	ExportNotes, AcroForm:ExportFDF
File→DocumentInfo	GeneralInfo, OpenInfo, FontsInfo, SecurityInfo, Weblink:Base, AutoIndex:DocInfo
File→Preferences	GeneralPrefs, NotePrefs, FullScreenPrefs, Weblink:Prefs, AcroSearch:Preferences(Windows) or, AcroSearch:Prefs(Mac), Cpt:Capture
Edit	Undo, Cut, Copy, Paste, Clear, SelectAll, Ole:CopyFile, TouchUp:TextAttributes, TouchUp:FitTextToSelection, TouchUp:ShowLineMarkers, TouchUp:ShowCaptureSuspects, TouchUp:FindSuspect, Properties
Edit→Fields	AcroForm:Duplicate, AcroForm:TabOrder
Document	Cpt:CapturePages, AcroForm:Actions, CropPages, RotatePages, InsertPages, ExtractPages, ReplacePages, DeletePages, NewBookmark, SetBookmarkDest, CreateAllThumbs, DeleteAllThumbs
View	ActualSize, FitVisible, FitWidth, FitPage, ZoomTo, FullScreen, FirstPage, PrevPage, NextPage, LastPage, GoToPage, GoBack, GoForward, SinglePage, OneColumn, TwoColumns, ArticleThreads, PageOnly, ShowBookmarks, ShowThumbs
Tools	Hand, ZoomIn, ZoomOut, SelectText, SelectGraphics, Note, Link, Thread, AcroForm:Tool, Acro_Movie:MoviePlayer, TouchUp:TextTool, Find, FindAgain, FindNextNote, CreateNotesFile
Tools→Search	AcroSrch:Query, AcroSrch:Indexes, AcroSrch:Results, AcroSrch:Assist, AcroSrch:PrevDoc, AcroSrch:PrevHit, AcroSrch:NextHit, AcroSrch:NextDoc

Window	ShowHideToolBar, ShowHideMenuBar, ShowHideClipboard, Cascade, TileHorizontal, TileVertical, CloseAll
Help	HelpUserGuide, HelpTutorial, HelpExchange, HelpScan, HelpCapture, HelpPDFWriter, HelpDistiller, HelpSearch, HelpCatalog, HelpReader, Weblink:Home
Help(Windows)	About

9 PDF and HTML forms

You must put your fields inside a `Form` environment. The environment does some general setups, so should be used only once in a document. Using simply `\Form` at the begin of the document is possible too.

There are six macros to prepare fields:

`\TextField[`*parameters*`]{`*label*`}`

`\CheckBox[`*parameters*`]{`*label*`}`

`\ChoiceMenu[`*parameters*`]{`*label*`}{`*choices*`}`

`\PushButton[`*parameters*`]{`*label*`}`

`\Submit[`*parameters*`]{`*label*`}`

`\Reset[`*parameters*`]{`*label*`}`

The way forms and their labels are laid out is determined by:

`\LayoutTextField{`*label*`}{`*field*`}`

`\LayoutChoiceField{`*label*`}{`*field*`}`

`\LayoutCheckField{`*label*`}{`*field*`}`

These macros default to #1 #2

What is actually shown in the field is determined by:

`\MakeRadioField{`*width*`}{`*height*`}`

`\MakeCheckField{`*width*`}{`*height*`}`

`\MakeTextField{`*width*`}{`*height*`}`

`\MakeChoiceField{`*width*`}{`*height*`}`

`\MakeButtonField{`*text*`}`

These macros default to `\vbox to #2{\hbox to #1{\hfill}\vfill}`, except the last, which defaults to #1; it is used for buttons, and the special `\Submit` and `\Reset` macros.

You may also want to redefine the following macros:

```
\def\DefaultHeightofSubmit{12pt}
\def\DefaultWidthofSubmit{2cm}
\def\DefaultHeightofReset{12pt}
\def\DefaultWidthofReset{2cm}
\def\DefaultHeightofCheckBox{0.8\baselineskip}
\def\DefaultWidthofCheckBox{0.8\baselineskip}
\def\DefaultHeightofChoiceMenu{0.8\baselineskip}
\def\DefaultWidthofChoiceMenu{0.8\baselineskip}
\def\DefaultHeightofText{\baselineskip}
\def\DefaultHeightofTextMultiline{4\baselineskip}
\def\DefaultWidthofText{3cm}
```

9.1 Forms environment parameters

`action`	*URL*	The URL that will receive the form data if a Submit button is included in the form
`encoding`	*name*	The encoding for the string set to the URL; FDF-encoding is usual, and `html` is the only valid value
`method`	*name*	Used only when generating HTML; values can be `post` or `get`

9.2 Forms optional parameters

Note that all colors must be expressed as RGB triples, in the range 0..1 (i.e. `color=0 0 0.5`)

`accesskey`	key		(as per HTML)
`align`	number	*0*	alignment within text field; 0 is left-aligned, 1 is centered, 2 is right-aligned.
`altname`	name		alternative name, the name shown in the user interface
`backgroundcolor`			color of box
`bordercolor`			color of border

bordersep			box border gap
borderstyle	char	*S*	box border style; S (Solid) is default, B is Beveled, D is Dashed, I is Inset and U is Underline
borderwidth		*1*	width of box border, the value is a dimension or a number with default unit bp
calculate			JavaScript code to calculate the value of the field
charsize	dimen		font size of field text
checkboxsymbol	char	*4 (✔)*	symbol used for check boxes (ZapfDingbats), the value is a character or \ding{*number*}, see package pifont from bundle psnfss
checked	boolean	*false*	whether option selected by default
color			color of text in box
combo	boolean	*false*	choice list is 'combo' style
default			default value
disabled	boolean	*false*	field disabled
format			JavaScript code to format the field
height	dimen		height of field box
hidden	boolean	*false*	field hidden
keystroke			JavaScript code to control the keystrokes on entry
mappingname	name		the mapping name to be used when exporting the field data
maxlen	number	*0*	number of characters allowed in text field
menulength	number	*4*	number of elements shown in list
multiline	boolean	*false*	whether text box is multiline
name	name		name of field (defaults to label)
onblur			JavaScript code
onchange			JavaScript code
onclick			JavaScript code
ondblclick			JavaScript code
onfocus			JavaScript code
onkeydown			JavaScript code
onkeypress			JavaScript code
onkeyup			JavaScript code
onmousedown			JavaScript code
onmousemove			JavaScript code
onmouseout			JavaScript code
onmouseover			JavaScript code
onmouseup			JavaScript code
onselect			JavaScript code
password	boolean	*false*	text field is 'password' style
popdown	boolean	*false*	choice list is 'popdown' style
radio	boolean	*false*	choice list is 'radio' style
radiosymbol	char	*H (★)*	symbol used for radio fields (ZapfDingbats), the value is a character or \ding{*number*}, see package pifont from bundle psnfss
readonly	boolean	*false*	field is readonly
rotation	number	*0*	rotation of the widget annotation (degree, counterclockwise, multiple of 90)
tabkey			(as per HTML)
validate			JavaScript code to validate the entry
value			initial value
width	dimen		width of field box

10 Defining a new driver

A hyperref driver has to provide definitions for eight macros:

1. \hyper@anchor
2. \hyper@link
3. \hyper@linkfile
4. \hyper@linkurl
5. \hyper@anchorstart
6. \hyper@anchorend
7. \hyper@linkstart
8. \hyper@linkend

The draft option defines the macros as follows

```
\let\hyper@@anchor\@gobble
\gdef\hyper@link##1##2##3{##3}%
\def\hyper@linkurl##1##2{##1}%
\def\hyper@linkfile##1##2##3{##1}%
\let\hyper@anchorstart\@gobble
\let\hyper@anchorend\@empty
\let\hyper@linkstart\@gobbletwo
\let\hyper@linkend\@empty
```

11 Special support for other packages

Package hyperref aims to cooperate with other packages, but while the long term goal is to remove patches and to move link support into the kernel, classes and packages there are still several possible sources for conflict, such as

- Packages that manipulate the bibliographic mechanism. Peter William's harvard package is supported. However, the recommended package is either Patrick Daly's natbib package that has specific hyperref hooks to allow reliable interaction or the biblatex package. Both packages cover a very wide variety of layouts and citation styles, all of which work with hyperref.
- Packages that changes \label and \ref macros. Since LaTeX 2023-06-01 the kernel and hyperref/nameref use the same \label definition. \label has a hook for external packages. There should be no need for external packages and classes to redefine them.
- Packages that do anything serious with the index.
- Packages that do anything serious with sectioning commands and the toc.

The hyperref package is distributed with a variant of the xr, xr-hyper[6], which support crossdocument links using LaTeX's normal \label/\ref mechanisms.

11.1 Package Compatibility

Currently only package loading orders are available:

[6] It will be merged with the xr package soon

11.1.1 algorithm

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

11.1.2 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 redefined to use gather:

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

11.1.3 amsrefs

The documentation of amsrefs claims that the package must be loaded after hyperref (it is unclear if that is really true) so the recommended package loading order is:

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

11.1.4 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}
```

11.1.5 babel/magyar.ldf

The old version 2005/03/30 v1.4j will not work. You need at least version 1.5, maintained by Péter Szabó, see CTAN:language/hungarian/babel/.

11.1.6 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.

11.1.7 bibentry

Workaround:

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

\usepackage{bibentry}
\usepackage{hyperref}
```

```
\begin{document}

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

11.1.8 bigfoot

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

11.1.9 chappg

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

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

11.1.10 count1to

Package 'count1to' adds several \@addtoreset commands that confuse 'hyperref'. Therefore \theH<...> has to be fixed:

```
\usepackage{count1to}
\AtBeginDocument{% *after* \usepackage{count1to}
  \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}}%
}
```

11.1.11 dblaccnt

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

```
\usepackage{hyperref}
\usepackage{dblaccnt}
```

or see entry for vietnam.

11.1.12 easyeqn

Not compatible, breaks.

11.1.13 ellipsis

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

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

(this will lead to wrong ellipsis in the bookmarks, so \texorpdfstring is needed).

11.1.14 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.

11.1.15 endnotes

Unsupported.

11.1.16 foiltex

Update to version 2008/01/28 v2.1.4b: Since version 6.77a hyperref does not hack into `\@begindvi`, 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.

11.1.17 footnote

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

11.1.18 linguex

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

11.1.19 ltabptch

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

11.1.20 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.

11.1.21 minitoc-hyper

This package is obsolete, use the up-to-date original package minitoc instead.

11.1.22 multind

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

11.1.23 natbib

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

11.1.24 nomencl

Example for introducing links for the page numbers:

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

11.1.25 ntheorem-hyper

This package is obsolete, use the up-to-date original package ntheorem instead.
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.

11.1.26 prettyref

```
%%% example for prettyref %%%
\documentclass{article}
\usepackage{prettyref}
\usepackage{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 %%%
```

11.1.27 setspace

```
\usepackage{setspace}
 \usepackage{hyperref}
```

11.1.28 sidecap

Nothing special is needed anymore.

11.1.29 subfigure

The package is obsolete. Use either subfig or subcaption

11.1.30 titleref

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

11.1.31 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`.

11.1.32 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).

11.1.33 ucs/utf8x.def

Note: utf8 is now the default in LaTeX and ucs is no longer recommended.

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.

11.1.34 varioref

Most previous problems with varioref should be resolved. There is only an open issue regarding `\vrefformat` (https://github.com/latex3/hyperref/issues/225).

It is recommended to load varioref always with the option nospace, see the documentation.

```
\usepackage[nospace]{varioref}
```

11.1.35 verse

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

11.1.36 vietnam

```
% pd1enc.def should be loaded before package dblaccnt:
 \usepackage[PD1,OT1]{fontenc}
 \usepackage{vietnam}
 \usepackage{hyperref}
```

11.1.37 XeTeX

Default for the encoding of bookmarks is `pdfencoding=unicode`. That means the strings are always treated as unicode strings. If `auto` or `pdfdoc` is forced it applies only if the string restricts to the printable ASCII set, The reason is that the `\special` does not support PDFDocEncoding.

In older versions hyperref contained special conversion code from UTF-16BE back to UTF-8 in a number of places for xetex to avoid the xdvipdfmx warning

`Failed to convert input string to UTF16...`

This is no longer needed with a current xdvipdfmx, so this code has been removed. `\csname HyPsd@XeTeXBigCharstrue\endcsname` should no longer be used.

12 Limitations[7]

12.1 Wrapped/broken link support

Only few drivers support automatically wrapped/broken links, e.g. pdftex, dvipdfmx, 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 overfull `\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).

12.2 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. This can currently (2023) be avoided by using the PDF management and to load at least the new-or-1 module from latex-lab:

```
\DocumentMetadata{testphase=new-or-1}
```

12.3 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`.

13 Hints[8]

13.1 Spaces in option values

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

[7] This section moved from the README file, needs more integration into the manual
[8] This section moved from the README file, needs more integration into the manual

```
\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).

13.2 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}
  ```

- Scientic 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}
  ```

- Another 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 %%%
```

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

Getting rid of it:

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

13.4 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 case, try package hypcap that implements a method to circumvent the problem.

13.5 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}
```

13.6 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}
```

13.7 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}}
```

14 History and acknowledgments

The original authors of hyperbasics.tex and hypertex.sty, from which this package descends, are Tanmoy Bhattacharya and Thorsten Ohl. Package hyperref started as a simple port of their work to LaTeX 2ε standards, but eventually I rewrote nearly everything, because I didn't understand a lot of the original, and was only interested in getting it to work with LaTeX. I would like to thank Arthur Smith, Tanmoy Bhattacharya, Mark Doyle, Paul Ginsparg, David Carlisle, T. V. Raman and Leslie Lamport for comments, requests, thoughts and code to get the package into its first useable state. Various other people are mentioned at the point in the source where I had to change the code in later versions because of problems they found.

Tanmoy found a great many of the bugs, and (even better) often provided fixes, which has made the package more robust. The days spent on RevTeX are entirely due to him! The investigations of Bill Moss into the later versions including native PDF support uncovered a good many bugs, and his testing is appreciated. Hans Hagen provided a lot of insight into PDF.

Berthold Horn provided help, encouragement and sponsorship for the dvipsone and dviwindo drivers. Sergey Lesenko provided the changes needed for dvipdf, and Hàn Thế Thành supplied all the information needed for pdftex. Patrick Daly kindly updated his natbib package to allow easy integration with hyperref. Michael Mehlich's hyper package (developed in parallel with hyperref) showed me solutions for some problems. Hopefully the two packages will combine one day.

The forms creation section owes a great deal to: T. V. Raman, for encouragement, support and ideas; Thomas Merz, whose book *Web Publishing with Acrobat/PDF* provided crucial insights; D. P. Story, whose detailed article about pdfmarks and forms solved many practical problems; and Hans Hagen, who explained how to do it in pdftex.

Steve Peter recreated the manual source in July 2003 after it had been lost.

Especial extra thanks to David Carlisle for the backref module, the ps2pdf and dviwindo support, frequent general rewrites of my bad code, and for working on changes to the xr package to suit hyperref.

15 GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

15.1 Applicability and definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic

paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

15.2 Verbatim copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 15.3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

15.3 Copying in quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

15.4 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 15.2 and 15.3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

15.5 Combining documents

You may combine the Document with other documents released under this License, under the terms defined in section 15.4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

15.6 Collections of documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

15.7 Aggregation with independent works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 15.3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

15.8 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 15.4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 15.4) to Preserve its Title (section 15.1) will typically require changing the actual title.

15.9 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

15.10 Future revisions of this license

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Addendum: how to use this license for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.