Hypertext marks in LaTeX: the hyperref package

Sebastian Rahtz

June 1998

Contents

1 Introduction

The package derives from, and builds on, the work of the HyperTeX project, described at `http://xxx.lanl.gov/hypertex/`. 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 HyperTeX specification[1] 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">`

1. This is borrowed from an article by Arthur Smith.

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 URL's 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 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; at present, the following drivers are supported:

hypertex dvi processors conforming to the HyperTEX guidelines (i.e. xdvi, dvips (with the `-z` option) and OzTeX)

dvips produces `\special` commands tailored for dvips

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.

pdftex Han The Thanh's TEX variant which writes PDF directly

dviwindo Y&Y's Windows previewer

vtex MicroPress' HTML and PDF-producing TEX variants

Output from dvips or dvipsone must be processed using Acrobat Distiller to obtain a PDF file. The result is generally prefererable to that produced by using the 'hypertex' driver, and then processing with `dvips -z`, but the dvi file is not portable.

2 Implicit behaviour

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

```
\usepackage{hyperref}
```

in the document preamble. 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. Hopefully you will find that all cross-references work correctly as hypertext. 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.

3 Additional user macros

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

`\href{`*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.

`\hyperbaseurl{`*URL*`}`

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

`\hyperimage{`*image URL*`}`

The image referenced by the *URL* is inserted.

`\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*

`\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.

4 Acrobat-specific behaviour

If you want to access the meno 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

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.

As an example, the behaviour 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[pdftitle={A Perfect Day},colorlinks=false]{hyperref}`

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 behaviour and page size.

`draft`	boolean	*false*	all hypertext options are turned off
`debug`	boolean	*false*	extra diagnostic messages are printed in the log file
`a4paper`	boolean	*true*	sets paper size to 210mm × 297mm
`a5paper`	boolean	*false*	sets paper size to 148mm × 210mm
`b5paper`	boolean	*false*	sets paper size to 176mm × 250mm
`letterpaper`	boolean	*false*	sets paper size to 8.5in × 11in
`legalpaper`	boolean	*false*	sets paper size to 8.5in × 14in
`executivepaper`	boolean	*false*	sets paper size to 7.25in × 10.5in

5.2 Configuration options

`raiselinks`	boolean	*true*	In the hypertex driver, the height of links is normally calculcated 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	*false*	Allows link text to break across lines; since this cannot be accomodated in PDF, it is only set true by default if the pdftex driver is used. This makes links on multiple lines into different PDF links to the same target.
`pageanchor`	boolean	*true*	Determines whether every page is given an implicit anchor at the top left corner. If this is turned off, `\tableofcontents` will not contain hyperlinks.
`plainpages`	boolean	*true*	Forces page anchors to be named by the arabic form of the page number, rather than the formatted form.
`nesting`	boolean	*false*	Allows links to be nested; no drivers currently support this.

5.3 Backend drivers

If no driver is specified, the package defaults to loading the `hypertex` driver.

`pdftex`	boolean	Sets up hyperref for use with the pdftex program.
`nativepdf`	boolean	an alias for dvips
`pdfmark`	boolean	an alias for dvips
`dvips`	boolean	Sets up hyperref for use with the dvips driver.
`hypertex`	boolean	Sets up hyperref for use with the HyperTEX-compliant drivers.
`dviwindo`	boolean	Sets up hyperref for use with the dviwindo Windows previewer.
`dvipsone`	boolean	Sets up hyperref for use with the dvipsone driver.
`vtex`	boolean	Sets up hyperref for use with MicroPress' VTEX; the PDF and HTML backends are detected automatically.
`latex2html`	boolean	Redefines a few macros for compatibility with latex2html.
`ps2pdf`	boolean	Redefines a few macros for compatibility with Ghostscript's PDF writer, otherwise identical to dvips

Note that 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}
```

5.4 Extension options

`extension`	text		Set the file extension (eg `dvi`) which will be appended to file links created if you use the xr package.
`hyperfigures`	boolean		
`backref`	boolean	*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`.
`pagebackref`	boolean	*false*	Adds 'backlink' text to the end of each item in the bibliography, as a list of page numbers.
`hyperindex`	boolean	*false*	Makes the text of index entries into hyperlinks. Easily broken ...
`colorlinks`	boolean	*false*	Colours 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.
`linkcolor`	color	*red*	Color for normal internal links.

`anchorcolor`	color	*black*	Color for anchor text.
`citecolor`	color	*green*	Color for bibligraphical citations in text.
`filecolor`	color	*magenta*	Color for URLs which open local files.
`menucolor`	color	*red*	Color for Acrobat menu items.
`pagecolor`	color	*red*	Color for links to other pages.
`urlcolor`	color	*cyan*	Color for linked URLs.

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

5.5 PDF-specific display options

`bookmarks`	boolean	*false*	A set of Acrobat bookmarks are written, in a manner similar to the table of contents, requiring two passes of LaTeX. Some post-processing 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.
`bookmarksnumbered`	boolean	*false*	If Acrobat bookmarks are requested, include section numbers.
`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
`pagebordercolor`	RGB color	*1 1 0*	The color of the box around links to pages
`urlbordercolor`	RGB color	*0 1 1*	The color of the box around links to URLs
`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.

Note that the color of link borders can be specified *only* as 3 numbers in the range 0..1, giving an RGB color. You cannot use colors defined in TeX.

5.6 PDF display and information options

`baseurl`	URL		Sets the base URL of the PDF document
`pdfpagemode`	text	*None*	Determines how the file is opening in Acrobat; the possibilies are `None`, `UseThumbs` (show thumbails), `UseOutlines` (show bookmarks), and `FullScreen`. 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
`pdfproducer`	text		Sets the document information Producer field
`pdfkeywords`	text		Sets the document information Keywords field
`pdfview`	text	*FitBH*	Sets the default PDF 'view' for each link
`pdfstartpage`	text	*1*	Determines on which page the PDF file is opened.
`pdfstartview`	text	*FitB*	Set the startup page view
`pdfpagescrop`	n n n n		Sets the default PDF crop box for pages. This should be a set of four numbers

6 PDF and HTML forms

You must put your fields inside a `Form` environment (only one per file).

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*`}`

`\LayoutCheckboxField{`*label*`}{`*field*`}`

These macros default to `#1 #2`

What is actually shown in as 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\DefaultWidthofText{3cm}
```

6.1 Forms optional parameters

Note that all colors must be expressed as RGB triples, in the range 0..1 (ie `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.
`backgroundcolor`			color of box
`bordercolor`			color of border
`bordersep`			box border gap
`borderwidth`			width of box border
`charsize`	dimen		font size of field text
`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
`height`	dimen		height of field box
`hidden`	boolean	*false*	field hidden
`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
readonly	boolean	*false*	field is readonly
tabkey			(as per HTML)
value			initial value
width	dimen		width of field box

7 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
```

History and acknowledgements

The original authors of hyperbasics.tex and hypertex.sty, from which this package descends, are Tanmoy Bhattacharya (tanmoy@qcd.lanl.gov) and Thorsten Ohl (thorsten.ohl@physik.th-darmstadt.de). 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 (bmoss@math.clemson.edu) into the later versions including native PDF support uncovered a good many bugs, and his testing is appreciated. Hans Hagen (pragma@pi.net) 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 Han The Thanh 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.

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.