

Hypertext capabilities with pdfT_EX

Federico Garcia

Email federook@gmail.com
Website www.fedegarcia.net

Contents

1	Introduction	2
2	Immediate special effects of hyperref	2
3	Arbitrary cross references	3
4	External links	4
4.1	Cross referencing between files	4
4.2	Links to the web	5
5	Bookmarks	5
6	Text and T_EXt	6
6.1	α -expressions and H ₂ O	7
6.2	Alpha-expressions and water	7
6.3	α -expressions and H ₂ O	7
7	Post-it notes	7
8	Hypersetup and options	8
9	A Tip	9
10	Material not covered	9
11	Documentation of hyperref	10

1 Introduction

With the standardization of electronic publishing and sharing of manuscripts, a wealth of new technical resources and possibilities is open to authors, resources that go beyond habitual on-paper typographic uses, to involve the new everyday tool of the reader: the mouse click. The $\text{T}_{\text{E}}\text{X}$ world, as usual, has been quick in absorbing the new possibilities. In particular, Hàn Thê Thành's $\text{pdfT}_{\text{E}}\text{X}$ (and its derivative $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$), and Sebastian Rahtz's hyperref $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ package—both of them available in standard distributions of $\text{T}_{\text{E}}\text{X}$ and $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ —are especially successful implementations of the possibilities of the PDF format—by far the most standardized, and by now monopolic, electronic format.

This article is intended as a quick guide to the most immediately usable functions of these tools. It is not intended to replace the available documentation, that is much more precise and complete. In fact, my intention is to steer away from completeness. For example, the file `options.dvi` in the hyperref distribution lists all the options to the package. Well, this list, that contains no explanations or running text, is enough to fill more than two pages—there's well over 60 options. The main fact about them, however, is that most of them are completely meaningless to the average user, who probably won't understand them anyway (like in my case). In other words, only a small portion of the universe of possibilities of on-screen documents is directly relevant for the average author. But the documentation of these features (the relevant ones) is all too often obscured by the rest of them. The whole business appears more overwhelming than it actually is. Hopefully, this article will help this perceived situation. It is an exposition of the (relevant) extended possibilities of $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ in terms of the $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ we all know. In some cases this will actually imply a lie, a white one. A footnote will state so, without going into further details.

2 Immediate special effects of hyperref

When hyperref is loaded (for useful options, and tips on loading the package, see [Hypersetup and options](#)) in an otherwise normal $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ document, a number of things happen without further intervention:

- The items in the table of contents, the list of figures, and the list of tables,

will be links: when the reader clicks on them, the cursor will jump to the corresponding target.

- The superscript that calls for a footnote will be a link to the footnote itself.
- Bibliographical references through `\cite` will create links to the entries in the final bibliography list.
- All pairs of `\label`-`\ref` will also produce links (the result of a `\ref` leading by mouse click to the corresponding `\label`).

The last point merits a few notes. First of all, this is true of *all* elements capable of cross referencing in L^AT_EX: not only chapters and sections, but also to enumerate items, equations, and footnotes.

Example

After

```
\footnote{This is the footnote.}\label{note}

\begin{enumerate}
\item \label{item1} This is the first item.
\item \label{item2} This, the second.
\end{enumerate}
```

a `\ref` to any of these keys will produce the corresponding number as a link.

On the other hand, the same is true for all `\pageref`, that prints the number of the *page* where the referenced element appears (rather than the number of the element itself). With `hyperref`, this page number will be a link.

For chapters and sections, in addition, `hyperref` offers a third command in the family: `\nameref`. Instead of printing the chapter/section number, `\nameref` will typeset *the name* of the chapter/section. This is much better and elegant than using the number if the document is intended to be read on the screen.

3 Arbitrary cross references

`\label` and `\ref` function in connection with L^AT_EX counters. But on-screen reading is not limited to refer to things that have a number. Thus, `hyperref` offers

commands for the creation of cross references that are independent from counters. These are `\hypertarget` and `\hyperlink`, exact analogous of `\label` and `\ref` respectively.

Both of these commands, however, have a second argument:

```
\hypertarget{<key>}{<text>}           (anal. to \label)
\hyperlink{<key>}{<text>}             (anal. to \ref)
```

For `\hypertarget`, `<text>` is the destination of the user's click; for `\hyperlink`, it is the text of the link itself. The `<text>`, in both cases, is any L^AT_EX box.

Example

The following code would create a picture (file `logo.png`) that is the destination of a link:

```
\hypertarget{arbréf1}{\includegraphics{logo.png}}
```

The link itself, with the text 'see the logo,' would be created with

```
\hyperlink{arbréf1}{see the logo}
```

4 External links

4.1 Cross referencing between files

The links created by `\label-\ref` and by `\hypertarget-\hyperlink` work within a single file. Cross referencing *between* files is possible thanks to a third pair of commands provided by `hyperref`.

This time, however, the commands are less than analogous to the usual `\label-\ref`. They require not one, but two 'keys' (one of them called `<category>` in the syntax below). Why this is so has to do with PDF syntax, but the L^AT_EX user can think of it as a second key.

```
\hyperdef{<category>}{<key>}
\hyperref{<file>}{<category>}{<key>}{<text>}
```

Example

The following line sets up a destination in the present file (note the space at the end of the line):

`\hyperdef{xmpl}{dest}_%`

An external file would link to this destination (in the present file, `garcia1.pdf`) with `\hyperref{garcia1.pdf}{xmpl}{dest}{to the destination}`.

4.2 Links to the web

The other kind of common external link is to a webpage. The command `\url` takes one argument—the destination’s URL address—and creates a link to it (a click on it opens the system’s browser on the requested page). For example, `\url{www.tug.org}` leads to the TUG website.

A related kind of link is the ‘mailto’ link: with a click on it, the system opens the local email program and creates a new message to the indicated email address. This is done through the command:

`\href{mailto:⟨email address⟩}{⟨link’s text⟩}`

The email address below my name in this article is such a link, created essentially by `\href{mailto:federook@gmail.com}{federook@gmail.com}`.¹

5 Bookmarks

The `hyperref` package handles virtually every bookmark-related need the average user comes across. If instructed (see [Hypersetup and options](#)), the package will automatically compile the bookmarks panel from the items in the table of contents. This includes sections, chapters, etc., and also the usual L^AT_EX results of `\addtocontents` and related commands.

There are two things that are not so direct: first, how to get a bookmark that does *not* correspond to an item in the TOC. For this, `hyperref` offers

`\pdfbookmark[⟨level⟩]{⟨text⟩}{⟨key⟩}`

The `⟨level⟩` is 0 for chapters, 1 for sections, etc. `-1`, in fact, for `\parts`. The `⟨text⟩` is the text of the bookmark itself. The `⟨key⟩` doesn’t really matter to us (it’s another of those PDF-format-related requirements), except for the requirement that it should be unique.

1. This is a white lie. Extra handling is done in `pracjourn.cls` to add a subject line to the email message.

The destination of a bookmark created by `\pdfbookmark` is the exact place where the command is issued. The bookmark itself will be appended, in the bookmark panels, in the position where the command is issued *with respect to other chapters, sections, etc.*

Example

The “Dummy Bookmark” in the present document was created right here with

```
\pdfbookmark[2]{“Dummy Bookmark”}{"bmkey"}
```

As a result, it appears (in the bookmarks panel) right below the one for this section (“Bookmarks”) and before the one for the next (“Text and \TeX ”). The fact that it appears as a subitem of “Bookmarks” is due to the [2] argument (the *level*).

So, how to get bookmarks appended to *other* places in the panel? (I.e., how to create a bookmark whose destination comes after a section heading, but with the bookmark itself being placed before the one for that section in the bookmarks panel?)

This question is not simply a puzzle, but has a potentially useful application: the mapping of the list of figures or the list of tables (as well as the table of contents) in the bookmarks panel. This is the second thing that is not so direct with `hyperref` (or `pdfLATEX`, for that matter).

In fact, it is a relatively hard thing to achieve. I found a solution when writing `pittetd`, the electronic dissertations class at the University of Pittsburgh. I have the project of abstracting this part of the `pittetd` code and upload it to CTAN as a small, independent package, but for the moment interested readers can consult the documentation of the solution in `pittetd.dtx`.

6 Text and \TeX

Bookmarks are the main environment of another issue PDF writers should be aware of: they are made only of plain text, and cannot support what \LaTeX can put in—for example—section titles. For example, the next subsection:

6.1 α -expressions and H₂O

This heading looks right in the text, but the corresponding bookmark is wrong.

The alternative takes care of the bookmark:

6.2 Alpha-expressions and water

But the heading in the text is not satisfactory.

6.3 α -expressions and H₂O

The solution is to use `hyperref`'s command `\texorpdfstring`:

```
\texorpdfstring{\langle TEX version \rangle}{\langle plain-text version \rangle}
```

The result is a flexible expression that behaves like T_EX text (T_EXt) in L^AT_EX context, like plain text in PDF-related strings.

Example

The present subsection was created with:

```
\subsection{\texorpdfstring{\$ \alpha \$}{Alpha}-expressions  
and \texorpdfstring{H$ _2 $O}{water}}
```

This procedure is generally not needed with accents and commands with immediate expansion, including common logos and symbols. But `hyperref` cannot convert more advanced stuff—notably math mode—without help. It is in these cases that `\texorpdfstring` is useful.

7 Post-it notes

In page 4 above there is an expandable ‘post-it’ note. In my opinion, this easily overlooked resource of PDF offers an excellent alternative to the footnote when a document is intended to be read on the screen. `hyperref` does not support these notes, but their creation (through pdfT_EX primitives) is not hard:

```
\pdfannot width w height h depth d { /Subtype /Text  
/Contents (<text>)}
```

The three dimensions `w`, `h`, and `d` are all \LaTeX dimensions. But the one that is important is `h` (the height) because it determines where the note appears in relation to the text baseline. It is also a good idea use a `\quad` after it.

One thing to have in mind with post-it notes is that their exact behavior (color, size, when it opens, how it closes, etc.) is not very much standardized, and tends to change from version to version of Acrobat Reader. That said, the note can be of a different color if, between the brackets of `\pdfannot`, `/C [r g b]` is appended; a title for the note is determined with `/T (title)`, and the note can show up open by default with `/Open true`.

If post-it notes are used at all, the document has to be typeset by pdfTeX (rather than converted from DVI), because otherwise the primitive `\pdfannot` is not available. See also [Hypersetup and options](#).

8 Hypersetup and options

The `hyperref` package has so many options that there is a separate command to handle them, namely `\hypersetup`. Thus, options to the package can be specified either in the usual way (`\usepackage[options]...`), or in a separate line:

```
\hypersetup{option, option, ...}
```

Some of the many options of `hyperref` are of particular interest.

```
colorlinks
linkcolor=red
urlcolor=blue
citecolor=green
```

```
bookmarks
bookmarksopen
bookmarksnumbered=false
```

```
pdftitle="Hypertext capabilities with pdf $\text{\LaTeX}$ "
pdfauthor="Federico Garcia"
pdfkeywords="Bookmarks, links, PDF, LaTeX"
```

The last three are useful to setup the fields of the ‘Document Properties’ dialog box in Acrobat Reader. (Again, the behavior of this is not reliably standard from version to version.)

Another thing to have in mind is that `hyperref` has to know the exact way the PDF file is produced: whether `pdfTeX` is run on the document, or a separate program will convert the DVI—in which case `hyperref` needs to load the corresponding driver. This information is given to the package as an option (`'pdftex'` for `pdfTeX`, `'dvipdfm'` if this program is used, etc.). Since this information is also used by other packages—notably `graphicx`—it is customary to indicate this option as a general option *to the document class*, which will then pass it to any package that needs it.

9 A Tip

`hyperref` changes the internal mechanism of cross references. The change is truly magical, in the sense that the user, most of the time, has and needs to have no idea that a change occurred.

However, there is one case in which the change becomes relevant: when a document is ‘converted’ from plain into hyper, or backwards (in other words, when the `\usepackage{hyperref}` line is added or removed. The presence of auxiliary files created by one mechanism and used by the other *will* create quite unintelligible error messages.

Therefore a tip: make sure to delete all auxiliary files (`.aux`, `.toc`, `.lot`, `.lof`, ...) when you will run a document for the first time with `hyperref` (or, later, for the first time without it).

10 Material not covered

Two main things are not covered in this document: ‘hyper-bibliography,’ in which the entries in the bibliography list would link back to the citations in the text; and ‘hyper-index,’ in which the page numbers in the index are links to the corresponding pages.

These topics are not so easy to deal with in a general manner. In the main, support for this tends to be fragile, because of the myriad of styles for both

bibliography and index. As a result, dealing with them all requires proficiency with `BIBTEX` and `MakeIndex`. Given this proficiency, the file `hyperref.dtx` is a source of information on how to get things done.

In the case of bibliographies, a more immediate guide is the file `backref.dvi`, actually the documentation to the sub-package of the same name (by David Carlisle and Sebastian Rahtz). The file is part of the `doc` directory of `hyperref`.

11 Documentation of `hyperref`

This is a description of the PDF files in the `doc` directory of `hyperref`:

manual by Sebastian Rahtz is a reference guide to the package. Not very useful to learn, it contains often crucial information on details, important when you are doing something extraordinary.

paper by Heiko Oberdiek is a precise and complete explanation of many of the topics treated here, from the point of view of ‘how does it work?’ (rather than ‘how do I use it?’).

slides is the set of slides used by Heiko in his presentation of ‘paper.’ It is an illustration of the possibilities of using `hyperref`, `pdfTEX`, and the package `thumbpdf`. (Access to its source file would be an amazing help...)