

The hypdvips package

Hyperref extensions for use with dvips

Raimund Niedrist

raimund.niedrist@student.uibk.ac.at

2008/09/02 v1.02

Abstract

The `hypdvips` package fixes some problems when using `hyperref` with `dvips`. It also adds support for breaking links, file attachments, embedded documents and different types of GoTo-links. The cooperation of `hyperref` with `cleveref` is improved, which in addition allows an enhanced back-referencing system.

Contents

1	Introduction	4
2	Usage	4
3	Command list	6
3.1	<code>\attachfile</code>	6
3.2	<code>\bmstyle</code>	8
3.3	Color commands	8
3.4	<code>\embedfile</code>	8
3.5	<code>\evenboxesstring</code>	9
3.6	<code>\file</code>	9
3.7	<code>\goto</code>	10
3.8	<code>\gotoparent</code>	12
3.9	<code>\listofattachments</code>	13
3.10	<code>\odest</code>	13
3.11	<code>\openaction</code>	13
3.12	<code>\pagelabel</code>	14
3.13	<code>\runattachment</code>	14

Trademark Information

Adobe®, Distiller®, PostScript® and Reader® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Mac, Mac OS and QuickTime are trademarks of Apple Inc., registered in the U.S. and other countries. The **hypdvips** package documentation is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Inc.

Ghostscript® is a registered trademark of Artifex Software, Inc.

JavaScript™ is a trademark of Sun Microsystems, Inc. in the United States and other countries.

List of Figures

1	Open-message shown with the Adobe Reader 6 software	5
	(a) standalone	5
	(b) in browser window	5
2	Sample <code>hypdvips</code> log file output	6
3	ID numbers of file attachments	11
4	Link to a child	11
5	Link to the parent	11
6	Link to a sibling	11
7	Link to an embedded file in an external document	12
8	Link from an embedded file to a normal file	12
9	Link to a grandchild	13
10	Link to a niece/nephew through the source's parent	13

List of Tables

1	Package options for <code>hypdvips</code>	5
2	Options for the <code>\attachfile</code> command	7
3	Options for the <code>\embedfile</code> command	9
4	Options for the <code>\goto</code> command	11

List of File Attachments

1	The MD5 Message-Digest Algorithm	
	Options: DETAILS, ATTACHMENTS TAB, SAVE A COPY..., LAUNCH ATTACHMENT	-
2	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types	
	Options: DETAILS, ATTACHMENTS TAB, SAVE A COPY..., LAUNCH ATTACHMENT	-
3	Generate MD5 hashes of files (with progress indicator)	
	Options: DETAILS, ATTACHMENTS TAB	6
4	Draft copy of this documentation with enabled <code>showdests</code> option	
	Options: DETAILS, ATTACHMENTS TAB	6
5	Bibliography file for <code>hypdvips.tex</code>	
	Options: DETAILS, ATTACHMENTS TAB, SAVE A COPY..., LAUNCH ATTACHMENT	-
6	Source code of this documentation	
	Options: DETAILS, ATTACHMENTS TAB	9

1 Introduction

The `hypdvips` package is a collection of fixes for problems when using `hyperref` with `dvips` as backend driver. As you may have noticed, the converted PostScript files created by the standalone `hyperref` package have some features missing compared to the direct PDF output of the `pdflatex` driver. The most severe deficit is probably the inability to break links. Another problem is, for example, that footnote links point to the baseline of the note, thus placing the footnote text itself outside of the reader window¹. Another issue is the linking to floats (e.g. a figure), where the link points to the baseline of the caption text, again leaving the figure outside of the window. There is a package that particularly addresses this issue (`hypcap`), but the original version doesn't work with breaking links.

In fact, every time I encountered a problem when using the `hyperref`/`dvips` bundle, I tried to fix it and put it into a collection. So, `hypdvips` has no specific purpose, I just thought it could be useful to share.

To make it short, the main features of `hypdvips` are:

- ▷ breaking links
- ▷ support for `cleveref` & `backref`
- ▷ file attachments
- ▷ embedded documents
- ▷ GoTo-, GoToR- & GoToE-links
- ▷ custom pagelabels
- ▷ document open-actions

2 Usage

The `hypdvips` package can be loaded using the following command:

```
\usepackage[options]{hypdvips}
```

Table 1 shows all possible package *options*. They can be specified using key/value pairs, e.g.

```
\usepackage[autotitle=true,JavaScript=false,Mac=true,showdests]{hypdvips}
```

If a boolean (true/false) option key is used without a specific value, it is assumed to be “true” (like `showdests` in the example above).

Currently, `hypdvips` applies changes to `hyperref`, `backref`, `hypcap`, `bookmark` & `cleveref`. All these packages (except for `cleveref`) are loaded implicitly by `hypdvips`, so there is no need to load them separately². If `cleveref` is used, it must be loaded **before** `hypdvips`.

PDF Reference links: This documentation often refers to the PDF Reference [1]. As the electronic form of the PDF Reference is a rather huge file (approx. 31 MB), it is not included in this documentation — but it is linked. To get these links to work, download the file http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf and put it into the same folder as this documentation.

¹ at least Adobe Reader software does so, but there may be other programs which put the link destination in the center of the window — in this case it wouldn't matter

² under certain circumstances `hyperref` must be loaded **before** `hypdvips`, especially if a `hyperref` option is used which can't be changed later using `\hypersetup`

Option	Default	Description
<code>autoauthor</code>	<code>false</code>	Automatically sets the <code>hyperref</code> option <code>pdfauthor</code> to the name given by the <code>\author</code> command.
<code>autocrop</code>	<code>true</code>	Automatically sets a cropbox for the PDF creator program — useful if it can't determine the paper size otherwise. The size of the cropbox is given by the <code>\paperwidth</code> and <code>\paperheight</code> commands.
<code>autotitle</code>	<code>false</code>	Automatically sets the <code>hyperref</code> option <code>pdftitle</code> to the title given by the <code>\title</code> command.
<code>detailedpagebr</code>	<code>true</code>	Only used in back-referencing: back-references that originally just point to the page top are expanded to point to the real occurrence of the citation on the page.
<code>evenboxes</code>	<code>false</code>	Sets the height of links to a fixed value. See the description of the <code>\evenboxesstring</code> command for further details.
<code>fish</code>	<code>true</code>	Only used when attaching files: Shows an animation on the console during calculation of MD5-checksums.
<code>fullbookmarks</code>	<code>false</code>	Creates bookmark entries for starred sections (e.g. the “Contents” or “References” sections).
<code>german</code>	<code>false</code>	Configures <code>hypdvips</code> to be used in german documents. This includes <code>cleveref</code> definitions and affects the List of File Attachments, the open-message and the back-referencing system ^a
<code>JavaScript</code>	<code>true</code>	Only used when attaching files: Allows or denies the document to contain JavaScript code. Most of the option buttons in the List of File Attachments won't work anymore when <code>JavaScript=false</code> .
<code>loabr</code>	<code>false</code>	Only used in back-referencing: Allows back-references to citations occurring in the List of File Attachments (e.g. if a file attachment description contains a citation).
<code>lofbr</code>	<code>false</code>	Only used in back-referencing: Allows back-references to citations occurring in the List of Figures (e.g. if a figure caption contains a citation).
<code>lotbr</code>	<code>false</code>	Only used in back-referencing: Allows back-references to citations occurring in the List of Tables (e.g. if a table caption contains a citation).
<code>Mac</code>	<code>false</code>	Only used when attaching files: Includes the resource fork of Mac files.
<code>openmessage</code>	<code>true</code>	Only used when attaching files: Shows a message concerning file attachments when opening the PDF document with some older Adobe Reader software versions. It instructs the user how to open the attached files (see fig. 1). The open-message is implemented with JavaScript code, so it won't work with <code>JavaScript=false</code> .
<code>quadpoints</code>	<code>true</code>	Only used for broken links: Uses a QuadPoints array to define the active link area. With <code>quadpoints=false</code> , independent links are created on each line. See table 8.24 and figure 8.9 of the PDF Reference [1] for further details.
<code>showdests</code>	<code>false</code>	Indicates horizontal link destinations by red lines — useful for hyperlink checking. Here you can see an embedded copy of this documentation compiled with <code>showdests=true</code> .
<code>tocbr</code>	<code>false</code>	Only used in back-referencing: Allows back-references to citations occurring in the Table of Contents (e.g. if a section heading contains a citation).
<code>twoside</code>	<code>false</code>	Sets the duplex parameter for the PDF creator program.

^a please [contact me](#) if you want to provide a translation into another language

Table 1: Package options for `hypdvips`

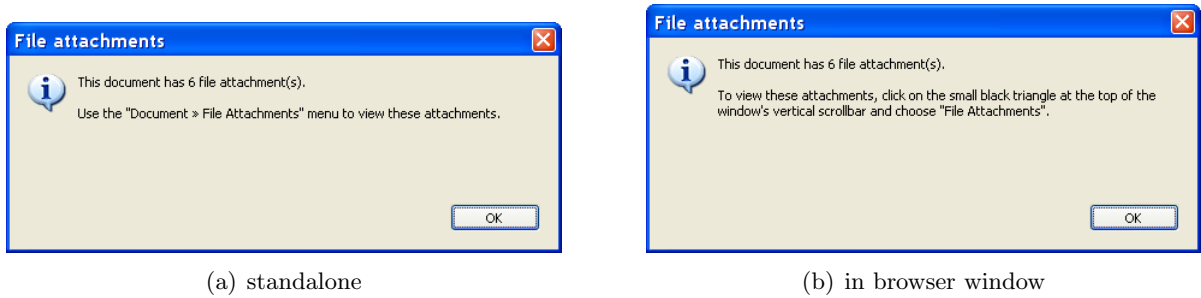


Figure 1: Open-message shown with the Adobe Reader 6 software

3 Command list

3.1 \attachfile[options]{filename}

Creates a PDF file attachment annotation, using data of the file referenced by *filename*. Table 2 shows all possible options. They are mostly similar to those of the \attachfile command of the attachfile/attachfile2 packages.

By default, the embedded file specification includes the size, MD5-checksum and creation/modification dates³ of the attached file. Author and subject fields of the annotation are also automatically filled in, if not otherwise specified by an option. For example, the command

```
\attachfile[author={Jem Berkes, SysDesign}, subject={MD5sums 1.2}, description={Generate MD5 hashes of files (with progress indicator)}, mimetype=application/zip, modified=, created=, color={0.2 0.65 1}, icon=tag]{c:/utils/md5sums-1.2.zip}
```

yields the following result⁴:

Another example (that creates the icon to the left):



```
\attachfile[description={Draft copy of this documentation with enabled {\ttfamily showdests} option}, name=hypdvips\string_showdests.pdf, mimetype=application/pdf, iconfilename=images/icon\string_draft.eps, rect={25 520 45 545}]{draft.pdf}
```

Filenames: Depending on the program used to convert the PostScript file to PDF, you have to specify the full path to the file or not. The Ghostscript application e.g. allows relative paths, but needs to be run with the `-dNOSAFER` command line argument if the attached file doesn't reside in the same directory as the PostScript file. The Adobe Distiller software always needs the full path, and besides from that, it has to be run without the `-F` command line argument for versions below 8.1, but **with** the `-F` command line argument for newer versions⁵.

File sizes: The **Size** entry in the embedded file parameter dictionary is limited to the highest supported integer number of the PDF creator program's PostScript interpreter. The usual limit of 32 bit-wide integers leads to a maximum size of 2,147,483,647 bytes (2 GB). At the beginning of processing, hypdvips logs information about the PostScript interpreter to the standard output file. There you can find whether 64 bit-wide integers are supported (see fig. 2).

```
GPL Ghostscript 8.62 (2008-02-29)
Copyright (C) 2008 Artifex Software, Inc. All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
Machine serial number: 42
Product name: GPL Ghostscript
Product revision level: 862
Interpreter version: 3010
LanguageLevel: 3
64-bit support: no

Calculating MD5-checksum of file: c:/documents/rfc/rfc1321.txt
<c6beb4140671d319f6433a3399cf6df2>

Calculating MD5-checksum of file: c:/documents/rfc/rfc2046.txt
<6225c196e3c5a20155f3a2082aea2801>





Calculating MD5-checksum of file: c:/utils/md5sums-1.2.zip
/7c57b00~c0cfbfbfbf~0cfb7c7c~20f097f~
```

Figure 2: Sample hypdvips log file output

³ if possible (depending on the PDF creator application and operating system used)

⁴ appearance may vary depending on the viewer application used

⁵ http://www.adobe.com/devnet/acrobat/downloads/Acrobat_SDK_readme.html#Known_Issues

Option	Default	Description
<code>anncreated</code>	(L ^A T _E X time)	Creation date of the annotation (CreationDate in annotation dictionary). ^a
<code>annmodified</code>	(L ^A T _E X time)	Modification date of the annotation (M in annotation dictionary). ^a
<code>author</code>	(value of <code>hyperref</code> option <code>pdfauthor</code>)	Name for the author field of the annotation (T in annotation dictionary).
<code>checksum</code>	(from file)	MD5-checksum of the file, written in hexadecimal format (Checksum in embedded file parameter dictionary; use <code>checksum=</code> to omit this entry). The MD5 algorithm is described in RFC 1321 [4].
<code>color</code>	(color of <code>\embeddedbordercolor</code>)	Color of the annotation icon, specified by three numbers between 0 and 1 according to the RGB color model (C in annotation dictionary).
<code>created</code>	(from file)	Creation date of the file (CreationDate in embedded file parameter dictionary). ^a
<code>creator</code>	(absent)	Only used with the package option <code>Mac</code> : The 4-character Mac OS file creator signature (Creator in Mac OS file information dictionary). E. g. <code>ogle</code> for QuickTime PictureViewer.
<code>description</code>	(absent)	Text for the description field of the annotation (Contents in annotation dictionary and Desc in embedded file specification dictionary).
<code>filetype</code>	(absent)	Only used with the package option <code>Mac</code> : The 4-character Mac OS file type (Subtype in Mac OS file information dictionary). E. g. <code>GIFf</code> for a Graphics Interchange Format (.GIF-)file.
<code>flags</code>	2#0000000100 (Print = true, all others false)	Flags for the annotation, written as PostScript integer (F in annotation dictionary). The default value e.g. can also be written decimal as 4 or hexadecimal as 16#4. Bit positions are: ^b 1 = Invisible, 2 = Hidden, 3 = Print, 4 = NoZoom, 5 = NoRotate, 6 = NoView, 7 = ReadOnly, 8 = Locked, 9 = ToggleNoView, 10 = LockedContents. See section 8.4.2 of the PDF Reference [1] for detailed meanings of these flags.
<code>icon</code>	<code>paperclip</code>	Name of the annotation icon (Name in annotation dictionary). Allowed names are:  <code>graph</code> ,  <code>pushpin</code> ,  <code>paperclip</code> and  <code>tag</code> . The actual appearance may vary depending on the viewer application used to read the PDF document.
<code>iconfilename</code>	(undefined, i.e. use <code>icon/text</code>)	Filename of an Encapsulated PostScript (.EPS-)file which acts as the annotation icon. This option must be paired with the <code>rect</code> option.
<code>mimetype</code>	(absent)	The MIME-type of the file (Subtype in embedded file stream dictionary; it can also be specified by a PDF first-class name, see appendix E of the PDF Reference [1] for details). MIME media types are specified in RFC 2046 [2]. An up-to-date list can be found at http://www.iana.org/assignments/media-types/ .
<code>modified</code>	(from file)	Modification date of the file (ModDate in embedded file parameter dictionary). ^a
<code>name</code>	<code>filename</code> (without path)	The filename for the F & UF entries of the file specification dictionary, written as PDF file specification string. Can be used to attach the file under a different name. See table 3.40 in section 3.10.1 of the PDF Reference [1] for information on how to write filenames of different operating systems.
<code>opacity</code>	1.0	The opacity of the annotation icon. Its value ranges from 0–1, where 0 means transparent and 1 means opaque.
<code>overprint</code>	<code>false</code>	Only used with the option <code>text</code> : Overprints the annotation text, thus making the annotation printable even if the “Print” flag is false. ^c Does not work when option <code>rect</code> is used.
<code>rect</code>	(rectangle at the current point, scaled to the current font size)	Set of 4 numbers which act as coordinates of a rectangle defining the position of the annotation icon/text. The first 2 numbers define the lower left, and the second 2 numbers the upper right corner of the rectangle. The numbers are given in default PDF coordinate space, where (0,0) is the lower left corner of the page with increasing values to the right and to the top. The resolution is 72 dots per inch.
<code>resourcefork</code>	<code>filename/.namedfork/rsrc</code>	Only used with the package option <code>Mac</code> : Filename of a file which holds the data for the resource fork of the embedded file stream (ResFork in Mac OS file information dictionary; use <code>resourcefork=</code> to omit this entry). ^d
<code>size</code>	(from file)	Size of the file, written as decimal number (Size in embedded file parameter dictionary; use <code>size=</code> to omit this entry)
<code>subject</code>	File attachment “name”	Text for the subject field of the annotation (Subj in annotation dictionary).
<code>text</code>	(undefined, i.e. use <code>icon/iconfilename</code>)	Uses any L ^A T _E X text to define the clickable area of the annotation. Can be used in conjunction with option <code>rect</code> to place the text anywhere on the page.
<code>timezone</code>	(absent)	The timezone offset which is appended to the L ^A T _E X time, written in the format <code>OHh'mm'</code> . ^a

^a see remark [Date format](#) in this section for details on how to write PDF dates

^b bit position 1 = LSB (least significant bit)

^c this is because Adobe Reader software seems to never print file attachment annotations, even if their “Print” flag is true

^d as I don’t own a Mac computer, I was unable to test whether the default value works to access the resource fork of the original file; this probably also depends on the PDF creator program used, so feedback on that is welcome!

Table 2: Options for the `\attachfile` command

Date format: PDF dates are written in the form `D:YYYYMMDDHHmmSSOHH'mm'`. `YYYY` is the year, `MM` is the month, `DD` is the day, `HH` is the hour, `mm` is the minute, `SS` is the second and `OHH'mm'` is the relationship of local time to Universal Time. `O` can be `+`, `-` or `Z` (=zero). The prefix `D:` and the apostrophe `'` characters in `OHH'mm'` are part of the syntax. See section 3.8.3 of the PDF Reference [1] for more information on PDF date strings.

See section 8.4.5 of the PDF Reference [1] for further information on file attachment annotations.

3.2 `\bmstyle{level}{style}`

Sets the appearance of a certain bookmark level. *level* can be a positive number or the character `*` (=bookmarks originating from starred sections). Valid *style* values are: `italic`, `bold` and `italic,bold`. The styles are cumulative, i.e. if a certain bookmark level is defined as bold and starred sections are defined as italic, then a starred section in this certain bookmark level will be italic & bold. E.g. the commands used in this documentation are:

```
\bmstyle{1}{bold}
\bmstyle*{}{italic}
```

3.3 Color commands

There are two types of commands which change the color of links:

```
\backrefcolor{color}   (Default: hyperref option citecolor)
\embeddedcolor{color}  (Default: hyperref option runcolor)
\footnotecolor{color}  (Default: hyperref option linkcolor)
```

change the color of the link **text** and are only used with the `hyperref` option `colorlinks=true`. The *color* must be known to the `xcolor` package, see section 2.5.2 of the `xcolor` package documentation [3] for an explanation of how to define colors.

The other ones

```
\backrefbordercolor{R G B} (Default: hyperref option citebordercolor)
\embeddedbordercolor{R G B} (Default: hyperref option runbordercolor)
\footnotebordercolor{R G B} (Default: hyperref option linkbordercolor)
```

set the color of the link **border** according to the RGB color model. Values for *R*, *G* & *B* range from 0–1 and are separated by spaces.

3.4 `\embedfile[options]{filename}`

Attaches the file referenced by the path *filename* as embedded file. Table 3 shows all possible options. E.g. this is the command which embeds the bibliography of this documentation:

```
\embedfile[description={Bibliography file for {\ttfamily hypdvips.tex}},
mimetype=text/plain]{bibdat.bib}
```

Option	Default	Description
<code>checksum</code>	(from file)	MD5-checksum of the file, written in hexadecimal format (Checksum in embedded file parameter dictionary; use <code>checksum=</code> to omit this entry). The MD5 algorithm is described in RFC 1321 [4].
<code>created</code>	(from file)	Creation date of the file (CreationDate in embedded file parameter dictionary) ^a
<code>creator</code>	(absent)	Only used with the package option <code>Mac</code> : The 4-character Mac OS file creator signature (Creator in Mac OS file information dictionary). E. g. <code>ogle</code> for QuickTime <code>PictureViewer</code> .
<code>description</code>	(absent)	Descriptive text associated with the file (Desc in embedded file specification dictionary).
<code>filetype</code>	(absent)	Only used with the package option <code>Mac</code> : The 4-character Mac OS file type (Subtype in Mac OS file information dictionary). E. g. <code>GIFf</code> for a Graphics Interchange Format (<code>.GIF</code> -)file.
<code>mimetype</code>	(absent)	The MIME-type of the file (Subtype in embedded file stream dictionary; it can also be specified by a PDF first-class name, see appendix E of the PDF Reference [1] for details). MIME media types are specified in RFC 2046 [2]. An up-to-date list can be found at http://www.iana.org/assignments/media-types/ .
<code>modified</code>	(from file)	Modification date of the file (ModDate in embedded file parameter dictionary) ^a
<code>name</code>	<code>filename</code> (without path)	The filename for the F & UF entries of the file specification dictionary, written as PDF file specification string. Can be used to embed the file under a different name. See table 3.40 in section 3.10.1 of the PDF Reference [1] for information on how to write filenames of different operating systems.
<code>resourcefork</code>	<code>filename/.namedfork/rsrc</code>	Only used with the package option <code>Mac</code> : Filename of a file which holds the data for the resource fork of the embedded file stream (ResFork in Mac OS file information dictionary; use <code>resourcefork=</code> to omit this entry) ^b
<code>size</code>	(from file)	Size of the file, written as decimal number (Size in embedded file parameter dictionary; use <code>size=</code> to omit this entry)

^a see remark [Date format](#) in section 3.1 for details on how to write PDF dates

^b as I don't own a Mac computer, I was unable to test whether the default value works to access the resource fork of the original file; this probably also depends on the PDF creator program used, so feedback on that is welcome!

Table 3: Options for the `\embedfile` command

3.5 `\evenboxesstring{text}`

Only used with the option `evenboxes`: Sets the height of links to the height of any *text*. E. g. after the command

```
\evenboxesstring{X}
```

all links are [as high as the character X](#). This command can be used in conjunction with the `hyperref` option `pdfborderstyle={/W 1 /S /U}` to produce underlined links, where the line is [always at the same height](#).

The default value is a string containing all alphanumeric characters plus some parentheses and a superscript, to be sure to not produce links which are too small in height (or depth).

3.6 `\file{filename}{description}`

This is basically a shortcut of the `\attachfile` command. It attaches the file referenced by the path *filename* using the given *description*, `overprint=true` with low opacity and default options apart from that. E. g. the command

```
\file{c:/latex/documents/hypdvips.tex}{Source code of this documentation}
```

has the following result: [hypdvips.tex](#)

The annotation text has the color defined with the command `\embeddedcolor`.

3.7 `\goto[options]{text}`

This command is used to create links inside or between PDF documents. It features the capabilities of the

- ▷ `\hyperlink{name}{text}`
- ▷ `\href{file:filename}{text}`
- ▷ `\href{gotoe:options}{text}`

commands, but offers an uniform and easy-to-use interface. You just have to specify the destination, and `\goto` decides which type of GoTo-link is created, depending on the *options* used. Links inside a document are called GoTo-links, links to external documents are GoToR-links and links to or between embedded files are called GoToE-links. Table 4 shows all possible *options* of the `\goto` command. E.g. the command

```
\goto[dest=section.3]{This is a link to the section ‘‘Command list’’}
```

[This is a link to the section “Command list”](#)

makes a GoTo-link to the name object⁶ `/section.3`. You can also specify explicit destinations:

```
\goto[dest={[0 /FitR 100 530 520 620]}]{This is a link which centers on the abstract}
```

[This is a link which centers on the abstract](#)

When linking to external documents, you may have to use byte strings as named destinations:

```
\goto[dest=(M13.9.20535.3Heading.Named.Destinations), filename=pdf\_reference\_1-7.pdf,  
newwindow]{This is a link to the section ‘‘Named Destinations’’ in the PDF Reference  
\cite{pdfref}}
```

[This is a link to the section “Named Destinations” in the PDF Reference \[1\]](#)

Note that the destination in this example includes parentheses (and), to distinguish the named destination **byte string** from a named destination **name object**. See section 8.2.1 of the PDF Reference [1] for more information on PDF destinations. As you also may have noticed, the border color of the external link is different than in the two examples before. External links have the color of `hyperref` options `filecolor` and `filebordercolor`, whereas local links use the `linkcolor/linkbordercolor`. Links to embedded files have the color defined by `\embeddedcolor` & `\embeddedbordercolor`.

Links to embedded documents can be accomplished with the `id` or `target` option. `id` is used when linking to embedded documents which have been attached with the `\attachfile`, `\file` or `\embedfile` commands in the the current L^AT_EX document, whereas the option `target` can go farther. The following examples are orientated on example 8.12 of the PDF Reference [1]:

1. `\goto[dest={Chapter 1}, id=1]{Link to a child}`
2. `\goto[dest={Chapter 1}, target=/R/P]{Link to the parent}`
3. `\goto[dest={Chapter 1}, target={/R /P /T << /R /C /N (Attachment 2) >> }]{Link to a sibling}`

⁶ this is the type of destinations created by `hyperref`

Option	Default	Description
dest	[0 /Fit]	The destination to jump to in the target. It can be either a named destination (specified by a name object or a byte string) or an explicit destination. Explicit destinations are written with squared brackets; see table 8.2 of the PDF Reference for the syntax. Byte strings are written with parentheses. Name objects have a preceding slash / character. If the argument neither represents a valid explicit destination nor is enclosed by parentheses or preceded by a slash, then it will be converted to a name object (if possible) or to a byte string ^a .
filename	(absent, i.e. use current document as target document)	Path to an external file which acts as target document for the link. This option cannot be used with the id option.
id	(absent, i.e. use current document as target document)	The ID of an attached file which acts as target document for the link. This ID can be found in the List of File Attachments: it is the number to the left of the file description (see fig. 3). ID's are only valid within the current document, thus this option cannot be paired with the filename option.
target	(absent, i.e. target document is the final target which holds the destination)	The content of a target dictionary which specifies the final target of the link relative to the target document (given by the id or filename option). See table 8.52 and example 8.12 in the PDF Reference [1] for entries in a target dictionary.

^a Ghostscript seems to convert byte strings to name objects when writing to PDF — weird, but the links work anyway (at least with Adobe Reader software)

Table 4: Options for the `\goto` command

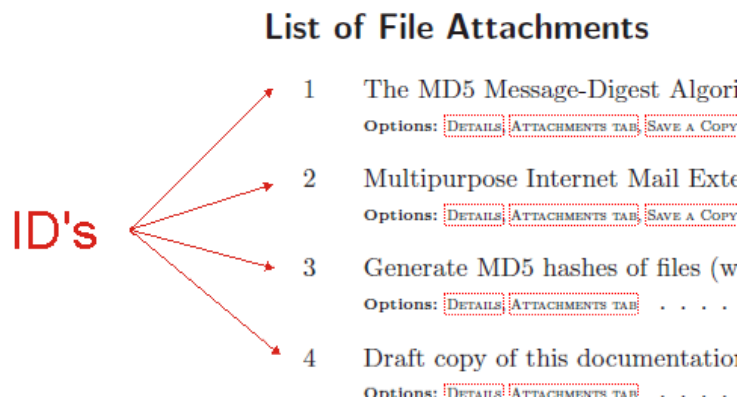


Figure 3: ID numbers of file attachments

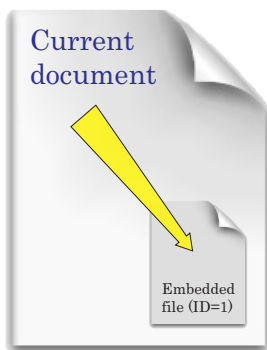


Figure 4: Link to a child

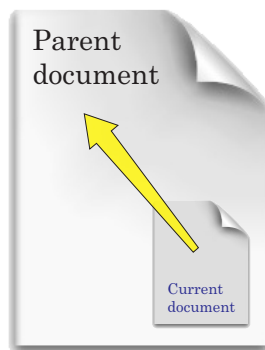


Figure 5: Link to the parent

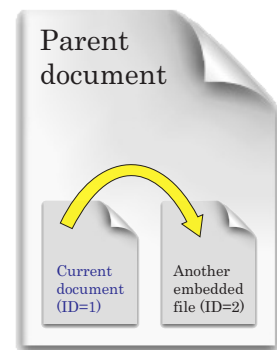


Figure 6: Link to a sibling

4. `\goto[dest={{Chapter 1}}, filename=someFile.pdf, target={/R /C /N (Attachment 1)}]{Link to an embedded file in an external document}`
5. `\goto[dest={{Chapter 1}}, filename=someFile.pdf]{Link from an embedded file to a normal file}`
6. `\goto[dest={{Chapter 1}}, id=1, target={/R /C /P 2 /A (Attachment 1)}]{Link to a grandchild}`
7. `\goto[dest=(destination), target={/R /P /T << /R /C /N (Attachment 2) /T << /R /C /P 3 /A (Attachment 1) >> >>}] {Link to a niece/nephew through the source's parent}`

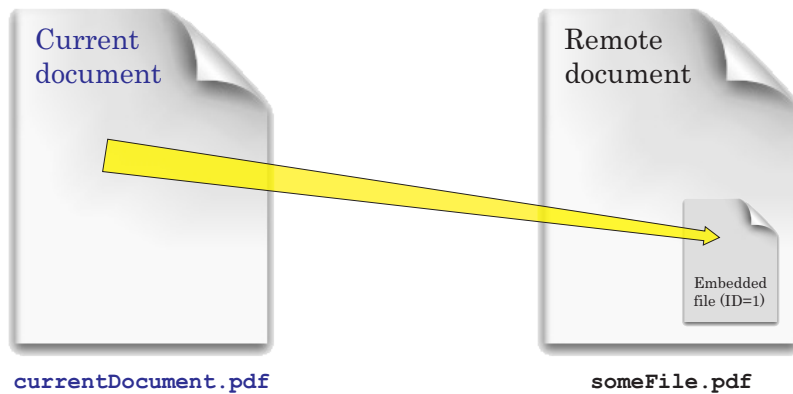


Figure 7: Link to an embedded file in an external document

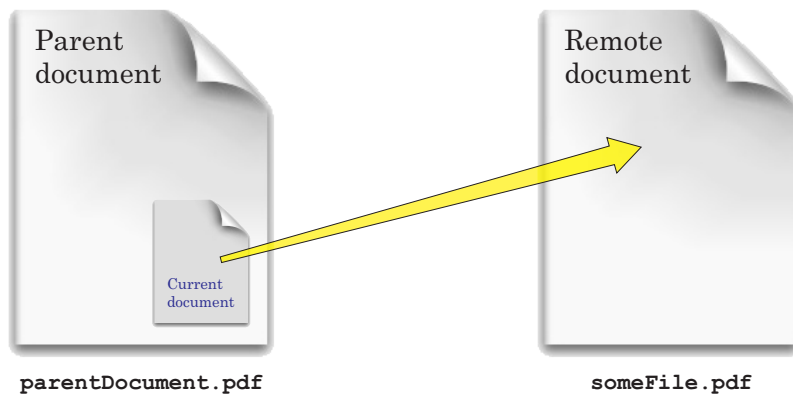


Figure 8: Link from an embedded file to a normal file

As the `id` option is just a shortcut, item 1. could also be written as

```
\goto[dest={{Chapter 1}}, target={/R /C /N (Attachment 1)}]{Link to a child}
```

or

```
\goto[dest={{Chapter 1}}, target={/R /C /P page /A (Attachment 1)}]{Link to a child}
```

depending on the type of the child (either embedded file or file attachment annotation).

3.8 `\gotoparent{destination}{text}`

This is a shortcut of the `\goto` command. It lets the given *text* point to a *destination* in the parent document.

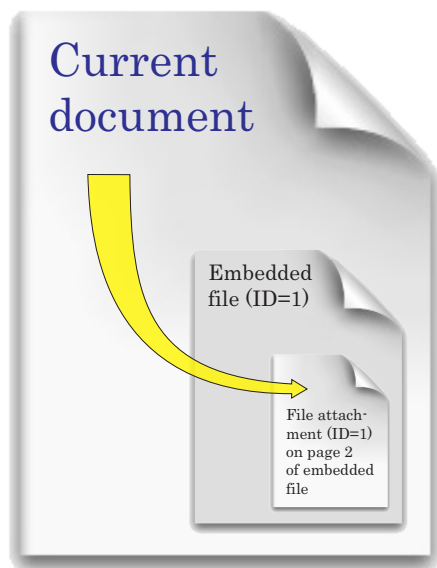


Figure 9: Link to a grandchild

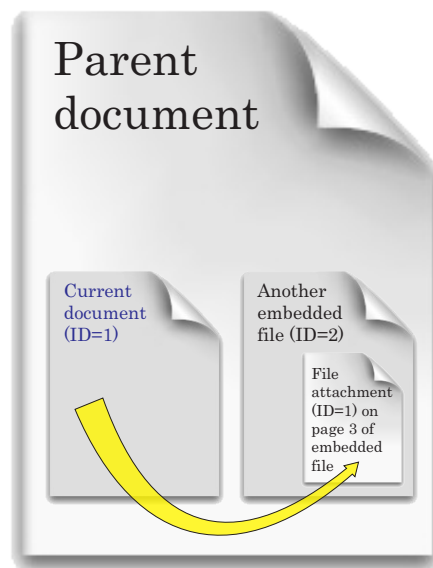


Figure 10: Link to a niece/nephew through the source's parent

3.9 `\listofattachments`

Creates a list of file attachments, analogous to the `\listoffigures` or `\listoftables` commands. The lines in this list are formatted by `\loaformat`, which can be changed to customize the appearance of the list:

```
\renewcommand{\loaformat}[4]{LATEX-code...}
```

The `\loaformat` command has 4 arguments:

- #1 = Attachment ID
- #2 = Attachment type (either `FileAttachment` or `EmbeddedFile`)
- #3 = *filename* from the corresponding `\attachfile` or `\embedfile` command
- #4 = all *options* that were given to the corresponding `\attachfile` or `\embedfile` command

3.10 `\odest{name}{offset}`

Creates the named destination `/name` located with a vertical *offset* relative to the current point. The *offset* can be given in any \LaTeX dimension. E.g. the command

```
\odest{odestexample}{1.5cm}
```

creates the destination `/odestexample` 1.5 cm above the `\odest` command. In the **draft copy with `showdests=true`** you can see the newly created destination. The `\hyperlink` command can be used to [link to that destination](#).

3.11 `\openaction{action}`

Sets the PDF document's open-action. *action* is the content of an action dictionary. E.g. this documentation uses an open-action to show the attachments tab:

```
\openaction{/N/ShowHideFileAttachment/S/Named}
```

See section 8.5 of the PDF Reference [1] for information on PDF actions and action dictionaries.

3.12 `\pagelabel[page]{pagelabel}`

Sets the PDF *pagelabel* for the specified *page*. The page number is optional — it defaults to the current page. E.g. the following command

```
\pagelabel{- \Roman{page}\space-}
```

uses the current page number in Roman format enclosed by dashes - as pagelabel for the current page.

Unnumbered pages: `hypdvips` modifies the `\thispagestyle` command to produce empty page-labels when the pagestyle is set to `empty`, as seen on the [title page](#) of this documentation.

3.13 `\runattachment{ID}{text}`

Creates a link from any *text* which launches the embedded file with the given *ID*. The color of the link can be defined with `\embeddedcolor` & `\embeddedbordercolor`. For example:

```
\runattachment{1}{Click here to open RFC 1321 \cite{rfc1321}}
```

[Click here to open RFC 1321 \[4\]](#)

Currently, the PDF JavaScript API only allows to export embedded files. Files in file attachment annotations can only be exported via the PDF viewer application. Note that with `JavaScript=false` the `\runattachment` command just produces the *text* without link.

References

- [1] Adobe Systems Incorporated. *PDF Reference*, sixth edition, November 2006. Adobe® Portable Document Format Version 1.7.

Referenced in: SECTION 2, TABLE 1, TABLE 2, TABLE 2, TABLE 2, SECTION 3.1, SECTION 3.1, TABLE 3, TABLE 3, SECTION 3.7, SECTION 3.7, SECTION 3.7, TABLE 4, SECTION 3.11

- [2] N. Freed and N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, November 1996. Updated by RFCs 2646, 3798 and 5147.
<http://tools.ietf.org/html/rfc2046>.

Referenced in: TABLE 2, TABLE 3

- [3] Dr. Uwe Kern. *Extending L^AT_EX's color facilities: the xcolor package*, January 2007.
<http://www.ctan.org/get/macros/latex/contrib/xcolor/xcolor.pdf>.

Referenced in: SECTION 3.3

- [4] R. Rivest. *The MD5 Message-Digest Algorithm*, April 1992.
<http://tools.ietf.org/html/rfc1321>.

Referenced in: TABLE 2, TABLE 3, SECTION 3.13