The hypdestopt package

Heiko Oberdiek*

2025-06-23 v2.9

Abstract

Package hypdestopt supports hyperref's pdftex and luatex driver. It removes unnecessary destinations and shortens the destination names or uses numbered destinations to get smaller PDF files.

The package is not compatible with the LaTeX PDFmanagement loaded with `\DocumentMetadata` and will do nothing if it detects it.

Contents

*Please report any issues at https://github.com/ho-tex/hypdestopt/issues

1 User interface

1.1 Introduction

Before PDF-1.5 annotations and destinations cannot be compressed. If the destination names are not needed for external use, the file size can be decreased by the following means:

- Unused destinations are removed.
- The destination names are shortened (option name).
- Using numbered destinations (option num).

1.2 Requirements

- Package hyperref 2006/06/01 v6.75a or newer ([2]).
- Package alphalph 2006/05/30 v1.4 or newer ([1]), if option name is used.
- Package iftex.
- pdfTeX 1.30.0 or newer.
- pdfTeX or luaTeX in PDF mode.
- ε-TeX extensions enabled.
- Probably an additional compile run of pdfLaTeX or luaLaTeX is necessary.

In the first compile runs you can get warnings such as:

```
! pdfTeX warning (dest): name{...} has been referenced ...
```

These warnings should vanish in later compile runs. However these warnings also can occur without this package. The package does not cure them, thus these warnings will remain, but the destination name can be different. In such cases test without the package, too.

1.3 Use

If the requirements are met, load the package:

```
\usepackage{hypdestopt}
```

The following options are supported:

verbose: Verbose debug output is enabled and written in the protocol file.

num: Numbered destinations are used. The file size is smaller, because names are no longer used. This is the default.

name: Destinations are identified by names.

1.4 Limitations

- Forget this package, if you need preserved destination names.
- Destination name strings use all bytes (0..255) except the carriage return (13), left parenthesis (40), right parenthesis (41), and backslash (92), because they must be quoted in general and therefore occupy two bytes instead of one.

  Further the zero byte (0) is avoided for programs that implement strings using zero terminated C strings. And 255 (0xFF) is avoided to get rid of a possible unicode marker at the begin.

  So far I have not seen problems with:

  - AcrobatReader 5.08/Linux
  - AcrobatReader 7.0/Linux
  - xpdf 3.00
  - Ghostscript 8.50
  - gv 3.5.8
  - GSview 4.6

  But I have not tested all and all possible PDF viewers.
- Use of named destinations (`\pdfdest`, `\pdfoutline`, `\pdfstartlink`, ...) that are not supported by this package.
- Currently only hyperref with pdfTeX and luaTeX in PDF mode is supported.

1.5 Future

A more general approach is a PDF postprocessor that takes a PDF file, performs some transformations and writes the result in a more optimized PDF file. Then it does not depend, how the original PDF file was generated and further improvements are easier to apply. For example, the destination names could be sorted: often used destination names would then be shorter than seldom used ones.

2 Implementation

2.1 Identification

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hypdestopt}%
4   [2025-06-23 v2.9 Hyperref destination optimizer (HO)]%
```

2.2 Check for PDF management

```
5 \IfPDFManagementActiveT
6  {
7    \PackageWarning{hypdestopt}
8     {
9      LaTeX PDF management detected. \MessageBreak
10     The package will do nothing.
11    }{}
12 }
```

```
\IfPDFManagementActiveT
 {
   \endinput
 }
```

2.3 Options

2.3.1 Option verbose

```
\newif\ifHypDest@Verbose
\DeclareOption{verbose}{\HypDest@Verbosetrue}
```

`\HypDest@VerboseInfo` Wrapper for verbose messages.

```
\def\HypDest@VerboseInfo#1{%
  \ifHypDest@Verbose
    \PackageInfo{hypdestopt}{#1}%
  \fi
}
```

2.3.2 Options num and name

The options num or name specify the method, how destinations are referenced (by name or number). Default is option num.

```
\newif\ifHypDest@name
\DeclareOption{num}{\HypDest@namefalse}
\DeclareOption{name}{\HypDest@nametrue}
\ProcessOptions*\relax
```

2.4 Check requirements

First pdfTeX must running in PDF mode.

```
\RequirePackage{iftex}[2019/11/07]
\RequirePackage{pdftexcmds}[2007/11/11]
\ifpdf
\else
  \PackageError{hypdestopt}{%
    This package requires pdfTeX or luaTeX in PDF mode%
  }\@ehc
  \expandafter\endinput
\fi
```

The version of pdfTeX must not be too old, because `\pdfescapehex` and `\pdfunescapehex` are used.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@escapehex\endcsname\relax
  \PackageError{hypdestopt}{%
    This pdfTeX is too old, at least 1.30.0 is required%
  }\@ehc
  \expandafter\endinput
\fi
```

Features of ε-TeX are used, e.g. `\numexpr`.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname numexpr\endcsname\relax
  \PackageError{hypdestopt}{%
    e-TeX features are missing%
  }\@ehc
  \expandafter\endinput
\fi
```

Package alphalph provides `\newalphalph` since version 2006/05/30 v1.4.

```
\ifHypDest@name
  \RequirePackage{alphalph}[2006/05/30]%
\fi
```

```
\RequirePackage{auxhook}[2009/12/14]
\RequirePackage{pdfescape}[2007/04/21]
```

2.5 Preamble for auxiliary file

Provide dummy definitions for the macros that are used in the auxiliary files. If the package is used no longer, then these commands will not generate errors.

\HypDest@PrependDocument We add our stuff in front of the `\AtBeginDocument` hook to ensure that we are before hyperref's stuff. Starting with LaTeX 2020/10/01 we use the hook and a rule.

```
\providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
\IfFormatAtLeastTF{2020/10/01}
  {
   \long\def\HypDest@PrependDocument#1{%
     \AddToHook{begindocument}{#1}}%
   \DeclareHookRule{begindocument}{.}{before}{hyperref}
  }
  {
   \long\def\HypDest@PrependDocument#1{%
     \begingroup
      \toks\z@{#1}%
      \toks\tw@\expandafter{\@begindocumenthook}%
      \xdef\@begindocumenthook{\the\toks\z@\the\toks\tw@}%
     \endgroup
    }
  }

\AddLineBeginAux{%
  \string\providecommand{\string\HypDest@Use}[1]{}%
}
```

2.6 Generation of destination names

Counter `HypDest` is used for identifying destinations.

```
\newcounter{HypDest}

\ifHypDest@name
```

\HypDest@HexChar Destination names are generated by automatically numbering with the help of package alphalph. `\HypDest@HexChar` converts a number of the range 1 until 252 into the hexadecimal representation of the string character.

```
  \def\HypDest@HexChar#1{%
    \ifcase#1\or
```

Avoid zero byte because of C strings in PDF viewer applications.

```
      01\or 02\or 03\or 04\or 05\or 06\or 07\or
```

Omit carriage return (13/`^^0d`). It needs quoting, otherwise it would be converted to line feed (10/`^^0a`).

```
      08\or 09\or 0A\or 0B\or 0C\or 0E\or 0F\or
      10\or 11\or 12\or 13\or 14\or 15\or 16\or 17\or
      18\or 19\or 1A\or 1B\or 1C\or 1D\or 1E\or 1F\or
      20\or 21\or 22\or 23\or 24\or 25\or 26\or 27\or
```

Omit left and right parentheses (40/`^^28`, 41/`^^39`), they need quoting in general.

```
      2A\or 2B\or 2C\or 2D\or 2E\or 2F\or
      30\or 31\or 32\or 33\or 34\or 35\or 36\or 37\or
      38\or 39\or 3A\or 3B\or 3C\or 3D\or 3E\or 3F\or
      40\or 41\or 42\or 43\or 44\or 45\or 46\or 47\or
      48\or 49\or 4A\or 4B\or 4C\or 4D\or 4E\or 4F\or
      50\or 51\or 52\or 53\or 54\or 55\or 56\or 57\or
```

Omit backslash (92/^^5C), it needs quoting.

```
      58\or 59\or 5A\or 5B\or 5D\or 5E\or 5F\or
      60\or 61\or 62\or 63\or 64\or 65\or 66\or 67\or
      68\or 69\or 6A\or 6B\or 6C\or 6D\or 6E\or 6F\or
      70\or 71\or 72\or 73\or 74\or 75\or 76\or 77\or
      78\or 79\or 7A\or 7B\or 7C\or 7D\or 7E\or 7F\or
      80\or 81\or 82\or 83\or 84\or 85\or 86\or 87\or
      88\or 89\or 8A\or 8B\or 8C\or 8D\or 8E\or 8F\or
      90\or 91\or 92\or 93\or 94\or 95\or 96\or 97\or
      98\or 99\or 9A\or 9B\or 9C\or 9D\or 9E\or 9F\or
      A0\or A1\or A2\or A3\or A4\or A5\or A6\or A7\or
      A8\or A9\or AA\or AB\or AC\or AD\or AE\or AF\or
      B0\or B1\or B2\or B3\or B4\or B5\or B6\or B7\or
      B8\or B9\or BA\or BB\or BC\or BD\or BE\or BF\or
      C0\or C1\or C2\or C3\or C4\or C5\or C6\or C7\or
      C8\or C9\or CA\or CB\or CC\or CD\or CE\or CF\or
      D0\or D1\or D2\or D3\or D4\or D5\or D6\or D7\or
      D8\or D9\or DA\or DB\or DC\or DD\or DE\or DF\or
      E0\or E1\or E2\or E3\or E4\or E5\or E6\or E7\or
      E8\or E9\or EA\or EB\or EC\or ED\or EE\or EF\or
      F0\or F1\or F2\or F3\or F4\or F5\or F6\or F7\or
```

Avoid 255 (0xFF) to get rid of a possible unicode marker at the begin of the string.

```
      F8\or F9\or FA\or FB\or FC\or FD\or FE%
    \fi
  }%
```

`HypDest@HexString` Now package alphalph comes into play. `\HypDest@HexString` is defined and converts a positive number into a string, given in hexadecimal representation.

```
  \newalphalph\HypDest@HexString\HypDest@HexChar{250}%
```

`\theHypDest` For use, the hexadecimal string is converted back.

```
  \renewcommand*{\theHypDest}{%
    \pdf@unescapehex{\HypDest@HexString{\value{HypDest}}}%
  }%
```

With option num we use the number directly.

```
\else
  \renewcommand*{\theHypDest}{%
    \number\value{HypDest}%
  }%
\fi
```

2.7 Assign destination names

`\HypDest@Prefix` The new destination names are remembered in macros whose names start with prefix `\HypDest@Prefix`.

```
\edef\HypDest@Prefix{HypDest\string:}
```

`\HypDest@Use` During the first read of the auxiliary files, the used destinations get fresh generated short destination names. Also for the old destination names we use the hexadecimal representation. That avoid problems with arbitrary names.

```
\def\HypDest@Use#1{%
  \begingroup
    \edef\x{%
      \expandafter\noexpand
      \csname\HypDest@Prefix\pdf@unescapehex{#1}\endcsname
    }%
    \expandafter\ifx\x\relax
      \stepcounter{HypDest}%
      \expandafter\xdef\x{\theHypDest}%
```

```
      \let\on@line\@empty
      \ifHypDest@name
        \HypDest@VerboseInfo{%
          Use: (\pdf@unescapehex{#1}) -\string> %
          0x\pdf@escapehex{\x} (\number\value{HypDest})%
        }%
      \else
        \HypDest@VerboseInfo{%
          Use: (\pdf@unescapehex{#1}) -\string> num \x
        }%
      \fi
    \fi
  \endgroup
}
```

After the first `.aux` file processing the destination names are assigned and we can disable `\HypDest@Use`.

```
\AtBeginDocument{%
  \let\HypDest@Use\@gobble
}
```

`\HypDest@MarkUsed` Destinations that are actually used are marked by `\HypDest@MarkUsed`. `\nofiles` is respected.

```
\def\HypDest@MarkUsed#1{%
  \HypDest@VerboseInfo{%
    MarkUsed: (#1)%
  }%
  \if@filesw
    \immediate\write\@auxout{%
      \string\HypDest@Use{\pdf@escapehex{#1}}%
    }%
  \fi
}%
```

2.8 Redefinition of hyperref's hooks

Package hyperref can be loaded later, therefore we redefine hyperref's macros at `\begin{document}`.

```
\HypDest@PrependDocument{%
```

2.8.1 Destination setting

luatex compatibility

```
\ifx\pdfextension\@undefined\else
  \protected\def\pdfdest{\pdfextension dest }
\fi
```

```
  \ifHypDest@name
    \let\HypDest@Org@DestName\Hy@DestName
    \renewcommand*{\Hy@DestName}[2]{%
      \EdefUnescapeString\HypDest@temp{#1}%
      \@ifundefined{\HypDest@Prefix\HypDest@temp}{%
        \HypDest@VerboseInfo{%
          DestName: (\HypDest@temp) unused%
        }%
      }{%
        \HypDest@Org@DestName{%
          \csname\HypDest@Prefix\HypDest@temp\endcsname
        }{#2}%
        \HypDest@VerboseInfo{%
          DestName: (\HypDest@temp) %
          0x\pdf@escapehex{%
```

```
          \csname\HypDest@Prefix\HypDest@temp\endcsname
        }%
      }%
    }%
  }%
\else
  \renewcommand*{\Hy@DestName}[2]{%
    \EdefUnescapeString\HypDest@temp{#1}%
    \@ifundefined{\HypDest@Prefix\HypDest@temp}{%
      \HypDest@VerboseInfo{%
        DestName: (\HypDest@temp) unused%
      }%
    }{%
      \pdfdest num%
      \csname\HypDest@Prefix\HypDest@temp\endcsname#2\relax
      \HypDest@VerboseInfo{%
        DestName: (\HypDest@temp) %
        num \csname\HypDest@Prefix\HypDest@temp\endcsname
      }%
    }%
  }%
\fi
```

2.8.2 Links

```
\let\HypDest@Org@StartlinkName\Hy@StartlinkName
\ifHypDest@name
  \renewcommand*{\Hy@StartlinkName}[2]{%
    \HypDest@MarkUsed{#2}%
    \HypDest@Org@StartlinkName{#1}{%
      \@ifundefined{\HypDest@Prefix#2}{%
        #2%
      }{%
        \csname\HypDest@Prefix#2\endcsname
      }%
    }%
  }%
\else
  \renewcommand*{\Hy@StartlinkName}[2]{%
    \HypDest@MarkUsed{#2}%
    \@ifundefined{\HypDest@Prefix#2}{%
      \HypDest@Org@StartlinkName{#1}{#2}%
    }{%
      \pdfstartlink attr{#1}%
                    goto num\csname\HypDest@Prefix#2\endcsname
      \relax
    }%
  }%
\fi
```

2.8.3 Outlines of package **hyperref**

```
\let\HypDest@Org@OutlineName\Hy@OutlineName
\ifHypDest@name
  \renewcommand*{\Hy@OutlineName}[4]{%
    \HypDest@Org@OutlineName{#1}{%
      \@ifundefined{\HypDest@Prefix#2}{%
        #2%
      }{%
        \csname\HypDest@Prefix#2\endcsname
      }%
    }{#3}{#4}%
  }%
\else
```

```
    \renewcommand*{\Hy@OutlineName}[4]{%
      \@ifundefined{\HypDest@Prefix#2}{%
        \HypDest@Org@OutlineName{#1}{#2}{#3}{#4}%
      }{%
        \pdfoutline goto num\csname\HypDest@Prefix#2\endcsname
                     count#3{#4}%
      }%
    }%
  \fi
```

Because `\Hy@OutlineName` is called after the `.out` file is written in the previous run. Therefore we mark the destination earlier in `\@@writetorep`.

```
  \let\HypDest@Org@@writetorep\@@writetorep
  \renewcommand*{\@@writetorep}[5]{%
    \begingroup
      \edef\Hy@tempa{#5}%
      \ifx\Hy@tempa\Hy@bookmarkstype
        \HypDest@MarkUsed{#3}%
      \fi
    \endgroup
    \HypDest@Org@@writetorep{#1}{#2}{#3}{#4}{#5}%
  }%
```

2.8.4 Outlines of package bookmark

```
  \@ifpackageloaded{bookmark}{%
      \renewcommand*{\BKM@DefGotoNameAction}[2]{%
        \@ifundefined{\HypDest@Prefix#2}{%
          \edef#1{goto name{hypdestopt\string :unknown}}%
        }{%
          \ifHypDest@name
            \edef#1{goto name{\csname\HypDest@Prefix#2\endcsname}}%
          \else
            \edef#1{goto num\csname\HypDest@Prefix#2\endcsname}%
          \fi
        }%
      }%
      \def\BKM@HypDestOptHook{%
        \ifx\BKM@dest\@empty
        \else
          \ifx\BKM@gotor\@empty
            \HypDest@MarkUsed\BKM@dest
          \fi
        \fi
      }%
  }{}%
}
```

⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/hypdestopt.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/hypdestopt.pdf Documentation.

3.2 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain TEX:

[1] CTAN:pkg/hypdestopt

```
tex hypdestopt.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
hypdestopt.sty → tex/latex/oberdiek/hypdestopt.sty
hypdestopt.pdf → doc/latex/oberdiek/hypdestopt.pdf
hypdestopt.dtx → source/latex/oberdiek/hypdestopt.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.3 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.4 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{hypdestopt.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLATEX:

```
pdflatex hypdestopt.dtx
makeindex -s gind.ist hypdestopt.idx
pdflatex hypdestopt.dtx
makeindex -s gind.ist hypdestopt.idx
pdflatex hypdestopt.dtx
```

4 References

[1] Heiko Oberdiek: *The alphalph package*; 2006/05/30 v1.4; CTAN:pkg/alphalph.

[2] Sebastian Rahtz, Heiko Oberdiek: *The hyperref package*; 2006/06/01 v6.75a; CTAN:pkg/hyperref.

5 History

[2006/06/01 v1.0]

- First version.

[2006/06/01 v2.0]

- New method for referencing destinations by number; an idea proposed by Lars Hellström in the mailing list LATEX-L.
- Options name and num added.

[2007/11/11 v2.1]

- Use of package pdftexcmds for LuaTEX support.

[2008/08/08 v2.2]

- Support for package bookmark added.

[2011/05/13 v2.3]

- Fix for `\Hy@DestName` if the destination name contains special characters.
- Fix for option name and package bookmark.

[2016/05/16 v2.4]

- Documentation updates.

[2016/05/21 v2.5]

- LuaTeX compatibility

[2019/12/29 v2.6]

- use iftex package.

[2020-09-02 v2.7]

- Adapted to the new hook management of LaTeX (github issue 1)

[2024-08-07 v2.8]

- Disable the package if `\DocumentMetadata` is detected (github issue 2)

[2025-06-23 v2.9]

- Adapted test for PDF management.

6 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.