

# The `hyperref-generic` module

## A generic driver for `hyperref`

The L<sup>A</sup>T<sub>E</sub>X Project\*

Version 0.95i, released 2021-08-28

This module generates a generic driver for `hyperref` meant to be used with the new L<sup>A</sup>T<sub>E</sub>X PDF management code. It is loaded automatically if the PDF management code is active. The name of the driver will change after the testphase.

The generic driver can be used with `pdfflatex`, `lualatex`, `xelatex`, `latex` with `dvipdfmx`, `latex` with `dvips+ps2pdf`. `latex` with `dvips+distiller` could work too but is untested. `(x)dvipdfmx` will probably soon support `dvilualatex`, then this combination should work too.

The driver *requires* the new PDF management code, so documents wanting to use it should start like this

```
\RequirePackage{pdfmanagement-testphase} %loads the code
\DeclareDocumentMetadata % activates it
{
  %% options
  %% e.g. pdf version, backend:
  % pdfversion=1.7,
  % backend = dvipdfmx
}
```

The new driver tries to be compatible with the standard `hyperref` drivers but there are nevertheless differences. Some of them due to the still experimental status of the driver, others are design decisions: one part of the project is to clean up and modernize the code. The following sections try to describe the differences but also to document some of the rationales of the changes, and to add some details and comments about the existing options and so to extend the `hyperref` manual.

## 1 Avoiding transition problems

Some code will only work properly after other packages have been adapted to the new PDF management code and the changes in this driver. This will take some time. Until then it is recommended to follow the following rules

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases `hyperref` interferes. So it is recommended for most options —with the exception of a few mentioned below in section 9—to set them in `\hypersetup`, not as package option.

---

\*E-mail: [latex-team@latex-project.org](mailto:latex-team@latex-project.org)

- This driver uses the `l3color` module for the colors. All colors defined with `\color_set:nn` or `\color_set:nnn` will work. Colors defined with `xcolor` will work if they don't use one of the special color models not supported by `l3color` as `pdfmanagement-firstaid` contains a patch for `xcolor`. If the package `color` is used it is currently recommended to define colors after `hyperref`.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

## 2 Bookmarks / outlines

The new driver doesn't contain code to handle bookmarks/outlines. Instead it forces the loading of the `bookmark` package unless the package option `bookmarks=false` has been used. Currently `bookmark` is loaded at the end of the preamble so if commands from `bookmark` are needed in the preamble the document should load it manually. This is subject to change at some time in the future.

## 3 “Metadata”

“Metadata”, informations about the document, are stored in a PDF in two places: The `/Info` dictionary and the XMP-metadata. `hyperref` only handles the `/Info` dictionary. The XMP-metadata can be added with like `pdfx` and `hyperxmp`.

The `/Info` dictionary can be filled with arbitrary keys, but the PDF viewer typically care only about a few, like `/Author`, `/Title` and `/Keywords`. A number of `/Info` keys, like dates and the producer, are added automatically by the engines and backends. Some of them can only be removed with special commands, some not at all. But—with the exception of `/Producer` when using the `dvips` backend—they can be overwritten.

The current handling of the metadata is problematic:

- External package like `hyperxmp` wants to access them too and for this had to patch a number of internal `hyperref` commands—which is a problem if the internal commands change (as happens with this new driver)
- `hyperref` (and also `hyperxmp`) tries to deduce some datas from document commands like `\title` or `\author`—something that worked reasonably well when only some standard classes with well-known definitions of these command existed, but gets problematic with classes and packages which define more powerful commands knowing a variety of optional arguments to set authors and affiliations and title information.

To resolve some of this problem the driver will

- *Not* try deduce author and title from documents. They have to be set in `\hypersetup` with `pdfauthor` and `pdftitle`. It is recommended to separate more than one author by commas, and to hide commas inside braces if needed:

```
pdfauthor = {Bär, Peter Anteater, {Riley, the sloth}}
```

- All values of relevant keys (including keys from the `hyperxmp` package) will be stored in a Metadata container, and can be retrieved with `\GetDocumentProperties`.

```
\edef\my@pdfauthor{\GetDocumentProperties{hyperref/pdfauthor}}
```

If the key hasn't be set, the result is empty. This gives external packages a public and reliable access to the data.

- `pdflang` is deprecated. Instead `\DeclareDocumentMetadata` should be used:

```
\DeclareDocumentMetadata{lang=de-DE}
```

The value can be retrieved as `document/lang`.

## 4 Dates

`hyperref` has a few keys to set dates. They typically expect the date in “PDF” format: `D:YYYYMMDDhhmmss+01'00'`.

One should be aware that `hyperxmp` will sometimes overwrites such settings.

## 5 PDF page size (mediabox)

The standard `hyperref` driver contain code to set the PDF page size. There is no real justification why this is done by `hyperref` apart from the fact that `LATEX` itself doesn't do it and that the needed special code could be added to the backend drivers.

In the new driver this code is gone. The reason is not that it is difficult to set the `MediaBox`, actually it could be done with one line of code:

```
\pdfmanagement_add:nnn{Page}{MediaBox}
  {[0-0~\dim_to_decimal_in_bp:n{\paperwidth}~
   \dim_to_decimal_in_bp:n{\paperheight}}}
```

The problem is to know which value to use (with the memoir class e.g. `\stockwidth` should be used instead of `\paperwidth`), and detecting this not a `hyperref` task. Instead the packages which change these values should also set the PDF page size. Also there are too many actors here: `color/graphicsx`, `geometry`, the KOMA-classes, memoir, ... all try to set this.

So if the PDF page size is wrong: load one of the other packages setting it e.g. the `color` or the `graphicsx` package.

## 6 Commands to create “external” references

`hyperref` has three commands related to external references like URL and file: `\url`, `\nolinkurl` and `\href`. The first two take one argument, while the last has two: the url and some free text.

`\url` and `\href` create link annotations. `\url` creates always an URI type, `\href` creates URI, GoToR and Launch depending on the structure of the argument.

`\href` has to create a (in the PDF) valid url or file name from its first argument. `\url` has to create a (in the PDF) valid url from its only argument and has also to print this argument as url. `\nolinkurl` only prints the url.

For the printing `\url` and `\nolinkurl` rely on the `url` package and its `\Url` command. (Expandable) commands are expanded and special chars can also be input by commands but beside this no conversion is done: for all input `hyperref` basically assumes that

the input is already a valid percent encoded url or a valid file name. `hyperref` also doesn't extend or add protocols.

As nowadays everyone is used to copy and paste links with all sorts of unicode into a browser and they work the `hyperref` input is clearly rather restricted.

So the new driver tries to extend the input and print options. Both `\href` and `\url` can now be told to accept non-ascii url's and to convert them internally to percent encoding. It is possible to define a standard protocol and so to avoid to have to type it all the time.

But extending the *print* options for `\url` and `\nolinkurl` while still using the `url`-package is hard to impossible in pdfL<sup>A</sup>T<sub>E</sub>X due to the way the `url` package works. Some chars can be added with the help of `\UrlSpecial` (at the cost of warnings) but it doesn't work for every input and documenting and explaining all the edge cases is no joy. So instead the new driver offers here the option to use different commands to format the printed output. It must be noted that this disable the special "hyphenation" method of url's.

## 6.1 Special problem: links to files

When a file is linked with `\href` than normally it is added as URI link. The exceptions are PDF's: for them PDF has the special type `GoToR` which allows also to link to a destination or a special page.

After a number of tests with various PDF viewer established that non-ascii files names don't work at all with a simple file name specification `GoToR` links now use a full filespec dictionary. This works better, but still no every PDF viewer support this correctly. on various system.

The following can be used to test viewers. It assumes that a `test.pdf`, a `grüßpdf.pdf` and a `grüße.txt` are in the current folder.

```
test-ascii  
test grüßpdf.pdf  
test grüße.txt
```

## 6.2 Splits

`\href` tries to be clever and to detect from the argument if a url or a file link or a launch command should be created.

The rules are not trivial, and they make the code complicated. This detection also makes it more difficult to handle special cases like non-ascii input for the link types.

For this reason three new commands have been create:

- `\hrefurl` for standard urls (and non-pdf files)
- `\hrefpdf` for references to pdf files
- `\hrefrun` for launch links

The new commands don't use prefixes like `\href`. Their argument should be the real content.

### 6.3 Options

All `\href` commands and `\url` have an option argument for keyval syntax. It accepts the following keys. Not all keys make sense for all keys, but they don't error, they are silently ignored. The optional argument can currently not be used together with the `\urldef` command.

key	applicable commands	note
<code>urlencode</code>	<code>\hrefurl</code>	if set the code will convert the argument to percent encoding. This allows non-ascii input.
<code>protocol</code>	<code>\hrefurl</code> , <code>\url</code>	This sets a prefix/protocol that is added to the url, see below.
<code>format</code>	<code>\url</code>	a command used to format the printed text. It replaces the standard <code>\Url</code> . This can improve non-ascii typesetting at the cost of losing the special line breaking.
<code>destination</code>	<code>\href</code> , <code>\hrefpdf</code>	A destination name in the PDF
<code>page</code>	<code>\href</code> , <code>\hrefpdf</code>	destination page, default: 1
<code>pdfremotestartview</code>	<code>\href</code> , <code>\hrefpdf</code>	start view, default: Fit
<code>ismap</code>	<code>\href</code> , <code>\hrefurl</code>	see PDF reference
<code>afrelationship</code>	<code>\href</code> , <code>\hrefpdf</code>	Changes the <code>/AFRelationship</code> key of the filespec dictionary. The value should be a PDF name without the starting slash.
<code>run-parameter</code>	<code>\hreflaunch</code>	run parameter (see the PDF reference)
<code>nextactionraw</code>	various	puts a <code>/Next</code> entry in the action dictionary (see the PDF reference)

The first four keys can be set also in `\hypersetup` for all following commands in the current group through the keys `href/urlencode`, `href/protocol`, `href/destination`, `href/format`.

It is possible to define own url commands with specific options e.g. with

```
\NewDocumentCommand\myurl{0{}}{\url[protocol=https://,format=\textsc,#1]}
```

## 7 Link decorations: border, color, OCG-color, ...

Some main changes are

- The default colors have been changed.
- Citations have by default no special color, they are colored like other internal links. You can use `citecolor` and `citebordercolor` to assign them a special color. This color is not reset if you use `allcolors` or switch to another color scheme. If you want the colors to follow `linkcolor` again you should remove the label `hyp/cite` and/or `hyp/citeborder` from the hook `hyp/link/cite`.
- a number of color schemes have been predefined.

## 7.1 Background information

With the standard drivers `hyperref` allows either to color the link text, or to use a border around it. There is also a (rather unknown) option `frenchlinks` to use small caps for some links instead of colors.

The *link border* is a setting in the PDF annotation directory. It can be colored and styled (with the `<xxx>bordercolor`, `pdfborderstyle` and `pdfhighlight` keys), but the exact look depends on the PDF viewer. Such decorations are normally not printed.

The *link text* is colored with the standard color commands for text. Such a color is also printed, which is often not wanted. The printing can be avoided in PDF with so-called OCG-layers: They allow to add variants of a text along with instructions which variant should be used for viewing and which for printing. `hyperref` implements a rather simple version for links: The link text is put in a box and printed twice with different colors on different OCG layers. As boxes are used such links can't be broken. The package `ocgx2` implements a more sophisticated version which allows to use it for links broken over lines and pages.

`hyperref` has keys to set the color and border for `link`, `url`, `file`, `menu` and `run` types. They correspond to the PDF annotation types `GoTo`, `URI`, `GoToR`, `Named` and `Launch`. Beside this there is a `anchorcolor` which isn't used at all, and `citecolor` which is a semantical category and doesn't fit to the other types.

In the standard drivers the decoration options are more or less exclusive and global: One of the options (`colorlinks`, `ocgcolorlinks`, or `borders`) has to be chosen in the preamble and is then used for the whole document and all link types. Only colors and eventually the border style can be adjusted locally. But there is no technical reason for these restrictions: It is quite possible to change all these attributes at any time both by link type and locally. The restrictions of the current implementation can only be explained by the age of the code: `hyperref` has been created at a time when memory was small and the main drivers were `html` and `postscript` based.

While link colors have been traditionally more or less under the control of `hyperref`, the situation with other format options, like the font, is more complicated. The font in `\url` is for example determined by `\Urlfont`, a command from the `url` package. In the case of internal (`GoTo`) references packages like `cleveref` or `biblatex` or `glossaries` offer formatting options too. Formatting here is often connected to semantics: an acronym should use a different font than a citation. While `hyperref` could offer options here, it would probably only clash with package formatting. It is more sensible not to interfere here. For this reason the `frenchlinks` option has been dropped.

## 7.2 New Keys

Some of the existing keys have been extended to allow individual setting for the link types `link`, `url`, `file` `menu` and `run`:

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc
- Beside `pdfhighlight` there are also `linkhighlight`, `urlhighlight` etc
- Beside `pdfborderstyle` there are also `linkborderstyle`, `urlborderstyle` etc
- Beside `colorlinks` there are also `colorlink`, `colorurl` etc
- Beside `ocgcolorlinks` there are also `ocgcolorlink`, `ocgcolorurl`, etc TODO

- Beside `hidelinks` there are also `hidelink`, `hideurl`, etc
- `bordercolormodel` allows to set the model used in annotations, the allowed values are `rgb` or `cmk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmk` (4 numbers) in the `/C` key of an annotation, this is often ignored by pdf viewers and the colors can be wrong.
- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

`colorscheme` (*setup key*) The new key `colorscheme` allows to switch the colors (both for text and borders) with a key word. It takes one of the values `primary-colors` (the colors as `hyperref` uses normally), `phelype`, `daleif`, `szabolcsA`, `szabolcsB`, `tivv`, `julian`, `henryford`. The names refer to the authors in answers and comments in <https://tex.stackexchange.com/questions/525261/better-default-colors-for-hyperref-links>. The default is `phelype`.

### 7.3 Public interfaces

The `colorlinks` and `ocgcolorlinks` and related keys are using these booleans:

```
\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_colorrurl_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorrurl_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

They are both inserting hook code in the `pdfannot/link/<type>/begin` and `pdfannot/link/<type>/end` hooks. `<type>` is one of `GoTo`, `URI`, `GoToR`, `Named` or `Launch`. `colorlinks` uses the label `hyp/color`, and `ocgcolorlinks` the label `hyp/ocg`.

They both use the same color names: `hyp/color/link`, `hyp/color/url`, `hyp/color/file`, `hyp/color/run`, `hyp/color/menu`.

The cite colors uses the names `hyp/color/cite` and `hyp/color/citeborder`.

The border colors aren't saved in color names currently, but if the need would arise it would possible to change this.

### 7.4 Changed behaviour

`colorlinks` `colorlinks` will as before disable the `pdfborder`, but it is possible to use the key in the document at any time, or to reenale the border if wanted. Internally `colorlinks` & friends will no longer define/undefine `\Hy@colorlink`, but instead use the hooks provided by the `l3pdfannot` package.

Color keys accept the following input syntax:

```
model based      urlbordercolor = [rgb]{1,1,0}
color expression urlbordercolor = red!50!blue
command          urlbordercolor = \mycolor
```

where `\mycolor` should expand to one of the other two syntax variants.

**frenchlinks** The option `frenchlinks` does nothing at all.

**cite colors** As mentioned above the support for `citecolor` and `citebordercolor` has been reduced. A package like `hyperref` can't keep track of such semantic contexts like `cite`, `acronym`, `glossaries` and `special references` and maintain keys for them. The keys are not completely dropped as this would affect packages like `natbib`, but they have been separated and are no longer affected by group keys like `allcolors` but must be set individually instead.

**link margin** The driver sets a default link margin—this is identical to `pdftex` and `lualatex` driver, but a change for the `xetex` and `dvips` driver. The (undocumented) command `\setpdflinkmargin` does nothing. Use either the key `pdflinkmargin` or `\pdfannot_link_margin:n` to change the margin. See also the description in section 14 and in the `hyperref` manual.

## 8 PDF strings

`hyperref` uses a command called `\pdfstringdef` to convert text input into something that makes sense and is valid in a PDF string, e.g. in the bookmarks or in the info dictionary or as form field values.

As the handling of the outlines are delegated to the `bookmark` package, they will for now still use `\pdfstringdef`, but all other strings produced by this driver will use a new method based on the `expl3` commands `\text_purify:n` and `\str_set_convert:Nnnn`. For normal text it shouldn't matter, but a variety of commands and math are handled differently. Like with `\pdfstringdef` they are a number of ways to adjust the outcome of `\text_purify:n`. These are described in the `expl3` documentation `interface3.pdf`.

*The new method is under heavy development!*

Important differences here are

- *This new method requires that files are utf8-encoded* (at least if non-ascii chars are used in for PDF strings).
- All robust commands are currently removed, unless an equivalent has been declared.
- Currently the new method is much more silent: it doesn't warn like `hyperref` if it removes commands.

## 9 Package options from hyperref

The driver will process the package options at the end. But normally options should better be set with `\hypersetup` after the package has been loaded. This is also the case for options which normally don't work in `\hypersetup`. One option that currently doesn't work correctly as package option is `ocgcolorlinks`

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `destlabels` (destination names are taken from `\label` if possible)



- `encap`
- `hyperfigures` (according to the `hyperref` manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- `hyperfootnotes`
- `hyperindex`
- `implicit` (redefine `LATEX` internals)
- `nesting` unneeded key, see comment below in 14. At some time it will be either removed or extended (if some use can be found).
- `pagebackref`
- `pdfpagelabels` (set PDF page labels)
- `psdextra` this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

- `debug`, `verbose` (a boolean)
- `bookmarks` (a boolean)
- `plainpages`
- `draft`, `final`
- `hypertexnames`
- `naturalnames`
- `pageanchor`

Ignored options:

- All driver options like `pdftex`, `dvipdfmx`, ...
- `raiselinks` (only used in the `dviwind`, `textures` and `tex4ht` driver anyway)
- `frenchlinks`
- `setpagesize`
- `addtopdfcreator`

## 10 Disabling links

`hyperref` knows like many packages the options `draft` and `final`. With `hyperref` they can be used as package options or in the preamble in `\hypersetup` and disable links and anchors completely. The new driver passes the options also to the `bookmark` package if `bookmark` hasn't been loaded yet as bookmarks can't work properly if the anchors from `hyperref` are missing.

`link` (*setup key*) The `draft` option is a global option that can't be undone (at least not easily). So the new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference without link (this is naturally also possible with `\ref*{abc}`). This disables also all hooks of the link type, so the link is for example no longer colored. It also removes the implicit grouping of the content.

## 11 Draftmode

`pdftex` and other engines knows a `draftmode` which can be set with `\pdfdraftmode=1` and `hyperref` honors this in some places. The new driver ignores it, for example `pagelabels` are created in any case. With today's computer power there is not much to gain and it only complicates the code.

This should not be confused with the `draft` and `final` package options! They are still honored.

## 12 Dropped options

A number of options are ignored by this driver

**pdfversion** The `pdfversion` should be set in `\DeclareDocumentMetadata`

**setpagesize** The key is ignored and the PDF page size is not set. Load `color` or `graphicx` or use a class which sets the PDF page size.

**breaklinks** The option does nothing sensible anyway (apart from triggering a warning). Currently with `latex+dvips` links can't be broken. But there is work in progress to change this.

**unicode** This is always true.

**pdfa** If this option is set to true `hyperref` normally checks and sets a small number of requirements for the PDF standard PDF/A. The key is ignored with this driver. Instead the wanted standard should be declared in `\DeclareDocumentMetadata`:

```
\DeclareDocumentMetadata{pdfstandard=A-2b}
```

Currently `A-1b`, `A-2b`, `A-3b` can be set. The support for various requirements is still incomplete, but the parts that `hyperref` checked are implemented:

- The `/F` key is added to links and `Print` is activated, `Hidden`, `Invisible`, `NoView` are deactivated.
- `/NeedAppearances` is suppressed
- `Pushbuttons`, which use the action `/S/JavaScript` are suppressed.

- Resetbuttons, which use the action `/S/ResetForm` are suppressed.
- In widget annotations, the `/AA` dictionary is suppressed.

## 13 Destinations

Destinations (sometimes call anchors in the `hyperref` documentation) are the places a link jumped too. Unlike the name may suggest they don't described an exact location in the PDF. Instead a destination contains a reference to a page along with an instruction how to display this page. The normally used "`XYZ top left zoom`" for example instructs the viewer to show the page with the given *zoom* and the top left corner at the *top left* coordinates—which then gives the impression that there is an anchor at this position.

From these instructions two (`Fit` and `FitB`) don't take an argument. All others take one (`FitH`, `FitV`, `FitBH`, `FitBV`) or more (`XYZ`, `FitR`) arguments. These arguments are normally coordinates, `XYZ` takes also a zoom factor. The coordinates are absolute coordinates in `bp` relative to the lower left corner of the PDF.

With the primitive command `\pdfdest` of `pdftex` almost all instructions are created with a keyword only: The needed coordinate is calculated automatically from the location the `\pdfdest` command is issued. So to get a specific coordinate one has to move the command to the right place. E.g.

```
\AddToHookNext{shipout/background}
  {\put(0,-\pdfpageheight+100bp){\pdfdest name{destA} FitH\relax}}
```

Exceptions are the `XYZ` instruction, where `pdftex` accepts a keyword `zoom` followed by a zoom factor, and the `FitR` instruction which understands the keywords `width`, `height` and `depth` followed by a dimension, which is then used to calculate a rectangle relative to the current location. If no keywords are given the dimensions are taken from the surrounding box—which can also lead to zero sized areas.

The manual of `hyperref` gives a bit the impression as if this coordinates can be set manually by the user but as described above this is mostly wrong: It is for normal destination only possible with a dvi-backend like `dvips` which make use of `pdfmark.def`. `pdftex` and `luatex` can use manual coordinates only for `pdfstartview` and `pdfremotestartview`. As `dvips` was the first driver of `hyperref` the option `pdfview` was at first developed for it and then adapted to `pdftex`. But this had the effect that the handling of the option `pdfview` is inconsequent across the backend and engines: For example with `pdfview=FitH 100` `pdftex` ignores the number and calculates its own, while `dvips` sets the coordinate to the absolute 100. The zoom factor of `XYZ` is not supported by the `pdftex` driver at all, and `FitR` only partially.

The generic driver consolidate this but tries to stay compatible with the other drivers as far as possible. It also takes into account that `pdfview` and `pdfstartview` and `pdfremotestartview` have different requirements: While for the first relative coordinates are fine, for the two others absolute coordinates are more sensible.

`pdfview` (*setup key*)      So with this driver the options `pdfview`, `pdfstartview` and `pdfremotestartview`  
`pdfstartview` (*setup key*) take the following options:

- `pdfremotestartview` (*setup key*)
- `Fit`, `FitB`, `FitH`, `FitV`, `FitBH`, `FitBV` which can be followed by a positive integer (separated by a space) or the keyword `null`. The number can be gives as a *dimension expression* surrounded with `\hypercalcbp`. The driver redefines this command to use `\dim_to_decimal_in_bp:n`.

- `pdfview` will ignore the integer and any other arguments and calculate the expected coordinates as described above for `pdftex` with all supported engines and backends.
- `pdfstartview` and `pdfremotestartview` will pass the optional number or keyword after expansion as absolute coordinate. Missing numbers will be filled up with `null`.
- `XYZ`. This can be followed (separated by spaces) by up to three positive integers or keywords `null` which are then taken as *top left zoom* in this order. *zoom* is a factor, so e.g. 0.5 will give a scaling of 50%.
  - `pdfview` will use the last value as *zoom*, ignore all other values and calculate the expected coordinates as described above for `pdftex` with all supported engines and backends (this means it is possible to use `XYZ 2` to set a zoom of 200%, it is not necessary to fill in dummy values.)
  - `pdfstartview` and `pdfremotestartview` will pass the optional numbers or keyword after expansion as absolute coordinates and zoom. Missing numbers will be filled up with `null`.

This new behaviour is in part incompatible with previous handling with the `dvips` driver.

- `FitR`. If no argument (separated by spaces) follows then `pdfview` will use with `pdftex` and `luatex` the automatic calculation of the coordinates from the encompassing box. With `dvips` and `(x)dvipdfmx` it will fall back to `Fit`. `pdfstartview` and `pdfremotestartview` will fallback to `Fit` too.
 

If arguments (separated by spaces) follow they should be four numbers representing *left bottom right top*.

  - `pdfview` will use the values to calculate coordinates relative to the current location. So `0 -100 200 400` will give a “box” of width 200bp, height 400bp and depth 100dp that the destination should encompass. Missing numbers will be set to 0. But one should be aware that it is quite unpredictable how viewers which support `FitR` handles zero sizes.
  - `pdfstartview` and `pdfremotestartview` will pass the values as absolute coordinates.

### 13.1 Names of destinations

`hyperref` creates two types of destination names: For numbered structures (so when the anchor is set by `\refstepcounter`) it builds the name from the counter name and the `\theH...` representation: `<counter name>.\theH<counter name>`.

For unnumbered structures, e.g. starred chapters or anchors created with `\phantomsection` it uses names like `section*.<number>` and `chapter*.<number>`.

Typically the name of destination can be retrieved by setting a label, this works also with unnumbered sections. The anchor and also the page can be retrieve in an expandable way with the help of commands from the `refcount` package which is loaded by `hyperref`. For example with the following commands it is possible to use the label to create a bookmark:

```
\bookmark[dest=\getrefbykeydefault{label}{anchor}{Doc-Start}]{my bookmark}
\bookmark[dest=page.\getrefbykeydefault{label}{page}{Doc-Start}]{my bookmark}
```

If a `\HyperDestNameFilter` is defined, this must be added around the definition, so actually the full code has to look like this

```
\bookmark[dest=
\HyperDestNameFilter{\getrefbykeydefault{label}{anchor}{Doc-Start}}]{mysection}
```

To simplify this hyperref provides `\hyperget{anchor}{label}` and `\hyperget{pageanchor}{label}`

## 14 Assorted key descriptions

The following gives a few details to some keys that are perhaps not completely described in the manual, or are a bit different in this driver. The list is alphabetic.

**bookmarkstype** (*setup key*) This key takes as value the extension of a list like `toc` or `lof`. If this list uses `\addcontentsline` the content will be added to the bookmarks. The key can be use in `\hypersetup` and also in the middle of the document to switch the list.

**bordercolormodel** (*setup key*) With `bordercolormodel` the colormodel used in the `/C` key of the annotation array and in similar keys is set. It does not affect the text and graphics colors in the page stream. Possible choices are `rgb` (three numbers in the array) and `cmk` (four numbers). While the PDF reference allows four numbers, PDF readers don't necessarily handle this correctly, so the value can be wrong.

**destlabel** (*setup key*) This is a boolean key. Currently it must be set as package option. If set to true, the name of a destination is taken from a following `\label`, if there is one before the next destination command. This requires two compilations to get the correct coordinates in the destination. In the first compilation the alias name is recorded in the aux-file:

```
\hyper@newdestlabel{section.1.2}{sec:sec2}
```

The next compilation can then make use of it. The two-pass could be avoided in the future with a better labeling system, where the name if set earlier.

**extension** (*setup key*) This key sets an variable that has two purposes: It is used if file name has not extension, and it decides if the annotation is a URI or GoToR annotation. So

```
\hypersetup{extension=dvi}
\href{mwe1.pdf}{pdf}
\href{mwe2.dvi}{dvi}
\href{mwe3}{no ext}
```

will create

```
/Subtype/Link/A<</S/URI /URI(mwe1.pdf)>>
/Subtype/Link/A<</S/GoToR /F (mwe2.dvi)>>
/Subtype/Link/A<</S/GoToR /F (mwe3.dvi)>>
```

**nesting** (*setup key*) Typically PDF viewer can handle only GoToR annotations pointing to a PDF. So normally the default value `pdf` of this key should not be changed. This key is useless in PDF context. The boolean is only used in the code for anchors/destination where nesting doesn't make sense. It should not be changed.

**pdfborder** (*setup key*) This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should

**linkborder** (*setup key*)

**urlborder** (*setup key*)

**runborder** (*setup key*)

**menuborder** (*setup key*)

according to the reference set round corners, but PDF viewer seem to ignore it. The third number is the line width of the border. Settings done with `pdfborderstyle` should take precedence.

<code>pdfborderstyle</code> ( <i>setup key</i> )	The value of this key is the content of the BS dictionary. As an example		
<code>linkborderstyle</code> ( <i>setup key</i> )	<code>/Type/Border /W 1 /S/U /D[3 2]</code>		
<code>urlborderstyle</code> ( <i>setup key</i> )	<u>Key</u>	<u>Values</u>	<u>comment / example</u>
<code>fileborderstyle</code> ( <i>setup key</i> )	<code>/Type</code>	<code>/Border</code>	optional
<code>runborderstyle</code> ( <i>setup key</i> )	<code>/W</code>	<code>&lt;number&gt;</code>	Width of border line
<code>menuborderstyle</code> ( <i>setup key</i> )	<code>/S</code>	<code>/S</code>	solid (default)
		<code>/D</code>	dash pattern can be set with <code>/D</code>
		<code>/B</code>	beveled
		<code>/I</code>	inset
		<code>/U</code>	underline
	<code>/D</code>	<code>&lt;array of numbers&gt;</code>	dash pattern (lines/gaps) (default [3])
<code>pdfcreationdate</code> ( <i>setup key</i> )	Setting these keys is normally not needed. If they are used the values are stored di-		
<code>pdfmoddate</code> ( <i>setup key</i> )	rectly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> . The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> ( <i>setup key</i> )	As described in the <code>hyperref</code> manual the behaviour differs between the backends: with <code>dvips</code> it is possible to change links locally, <code>pdflatex</code> and <code>lualatex</code> work by page, with <code>dvipdfmx</code> the setting is global (and has to be done in the preamble).		
<code>pdflang</code> ( <i>setup key</i> )	The key will work, but it is recommended to the set the language in <code>\DeclareDocumentMetadata</code> instead.		

## File I

# hyperref-generic driver implementation

```

1 <*package>
2 <@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2021-08-28 v0.95i %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
7 \ExplSyntaxOn
8 \file_input:n {hyperref-colorschemes.def}
9 \ExplSyntaxOff

```

## 1 messages

Redirect the message name:

```

10 \ExplSyntaxOn
11 \prop_gput:Nnn \g_msg_module_name_prop { hyp }{ hyperref }

```

At first a message for the testing of the resource management

```

12 \msg_new:nnnn
13 { hyp }

```

```

14 { missing-resource-management }
15 { The-PDF-resource-management-is-required-for-this-hyperref-driver! }
16 {
17   Activate-it-with \\
18   \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
19   \tl_to_str:n{\DeclareDocumentMetadata{<options>}}\\
20   before~\tl_to_str:n{\documentclass}
21 }

```

The pdfversion should be set in \DeclareDocumentMetadata

```

22 \msg_new:nnnn
23 { hyp }
24 { pdfversion-disabled }
25 {
26   This-hyperref-driver-ignores-the-pdfversion-key!\\
27   Set-the-pdfversion-in~\token_to_str:N \DeclareDocumentMetadata
28 }
29 {
30   For example:\\
31   \tl_to_str:n
32   {
33     \DeclareDocumentMetadata { pdfversion=1.7 }
34   }
35 }

```

A generic message for ignored keys.

```

36 \msg_new:nnn
37 { hyp }
38 { key-dropped }
39 {
40   This-hyperref-driver-ignores-the-key~#1!\\
41   Please-check-the-documentation.
42 }

```

pdf/A messages for fields, this will probably be moved to an external package

```

43 \msg_new:nnn
44 { hyp }
45 { pdfa-no-push-button }
46 { PDF/A:~Push-button-with-JavaScript-is-prohibited }
47
48 \msg_new:nnn
49 { hyp }
50 { pdfa-no-reset-button }
51 { PDF/A:~Reset-action-is-prohibited }

```

pdf/A message for not allowed Named actions

```

52 \msg_new:nnn
53 { hyp }
54 { pdfa-no-named-action }
55 { PDF/A:~Named-action-#1-is-prohibited }

```

A message if the destination name is empty.

```

56 \msg_new:nnn
57 { hyp }
58 { empty-destination-name }
59 {

```

```

60     Empty-destination-name,\\
61         using~'#1'
62     }

```

A message if the destination check fails

```

63 \msg_new:nnn
64 { hyp }
65 { invalid-destination-value }
66 {
67     Invalid-value~'#1'~of~'#2' \\
68     is~replaced-by~'Fit'~\msg_line_context:.
69 }

```

Some options or values should not be used in older pdf versions

```

70 \msg_new:nnn
71 { hyp }
72 { ignore-deprecated-or-unknown-option-in-pdf-version }
73 {
74     Option~'#1'~is-unknown~or~deprecated~in\\
75     pdf~version~#2.~Ignored.
76 }
77 \msg_new:nnn
78 { hyp }
79 { ignore-deprecated-or-unknown-value-in-pdf-version }
80 {
81     Value~'#1'~is-unknown~or~deprecated~in\\
82     pdf~version~#2.~Ignored.
83 }
84 \msg_new:nnn
85 { hyp }
86 { replace-deprecated-or-unknown-value-in-pdf-version }
87 {
88     Value~'#1'~is-unknown~or~deprecated~in\\
89     pdf~version~#2. Value~'#3'~is used instead.
90 }

```

During development not all standard hyperref keys are known and the Hyp-handler needs to process some new keys unknown to him. This issues warnings for now:

```

91 \msg_new:nnn
92 { hyp }
93 { unknown-key }
94 {
95     unknown-key~#2~of~module~'#1'~set~to~'#3'.
96 }
97 \msg_new:nnn
98 { hyp }
99 { unknown-key-to-Hyp }
100 {
101     ignored-in-family~Hyp-unknown-key~#1.
102 }

```

There are a lot choice keys. This defines messages which shows the valid choices if a faulty one has been used:

```

103 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
104 \msg_new:nnn

```



```

105 { hyp }
106 { unknown-choice }
107 {
108   Value-’#3’~is~invalid~for~key~’#1’.\\
109   The~key~accepts~only~the~choices\\
110   \clist_map_function:nN { #2 }\_hyp_clist_display:n
111 }
112
113 \msg_new:nnn
114 { hyp }
115 { unknown-choice+empty }
116 {
117   Value-’#3’~is~invalid~for~key~’#1’.\\
118   The~key~accepts~only~the~choices\\
119   \clist_map_function:nN { #2 }\_hyp_clist_display:n
120   An~empty~value~removes~the~setting.
121 }
122
123 \msg_new:nnn
124 { hyp }
125 { no-bool }
126 {
127   Value-’#2’~is~invalid~for~key~’#1’.\\
128   The~key~accepts~only~the~choices\\
129   *~true\\
130   *~false \\
131   *~and~an~empty~value~which~removes~the~setting.\\
132   No~value~is~equivalent~to~using~’true’.
133 }

```

A message for creator and producer which can't be removed.

```

134 \msg_new:nnn
135 { hyp }
136 { empty-info-value }
137 {
138   Empty~value~for~key~#1.\\
139   This~isn't~honored~by~all~backends.
140 }

```

## 2 Variants

```

141 \cs_generate_variant:Nn\pdf_destination:nn {nf}
142 \cs_generate_variant:Nn\pdf_object_ref:n {e}

```

## 3 Overwriting/providing commands from hyperref

hyperref checks driver version, we need to suppress this during the development

```

143 \chardef\Hy@VersionChecked=1 %don't check the version!
144 %\cs_set_protected:Npn \PDF@SetupDoc{}
145 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

`\hypercalcbp` We define a better (expandable) version of `\hypercalcbp`

```

146 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

(End definition for \hypercalcbp. This function is documented on page ??.)

This command must be provided for now, but they are unused by the driver:

```
147 \providecommand\@pdfborder{}
148 \providecommand\@pdfborderstyle{}
149 \newcommand\OBJ@OCG@view {} % needed in hyperref
150 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark
```

The pdfversion should be set in \DeclareDocumentMetadata but we must copy it to the hyperref command:

```
151 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
152 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
153 \legacy_if:nT { Hy@setpdfversion }
154 {
155     \msg_warning:nn { hyp }{ pdfversion-disabled }
156 }
157 \Hy@DisableOption{pdfversion}
```

\Acrobatmenu should use the new internal link command

```
158 \RenewDocumentCommand \Acrobatmenu { m m }
159 {
160     \hyper@linknamed {#1} {#2}
161 }
```

\hypersetup should set the new keys. We can't also execute \kvsetkeys{Hyp} as this errors for example with colors. This means the driver has to provide new code for every key!

```
162 % TODO should go at some time ...
163 % \kv@set@family@handler{Hyp}
164 % { \msg_warning:nxx {hyp}{unknown-key-to-Hyp}{#1} }
165 \cs_set_protected:Npn \hypersetup #1
166 {
167     \%kvsetkeys{Hyp} {#1}
168     \keys_set:nn { hyp / setup }{ #1 }
169 }
170 % TODO for now unknown keys should only give warnings.
171 \keys_define:nn { hyp / setup }
172 {
173     unknown .code:n =
174     {
175         \msg_warning:nxxxx { hyp } { unknown-key }
176         { hyp~/~setup }{ \l_keys_key_str } { #1 }
177     }
178 }
```

Hyperref creates a number of destinations automatically. E.g. in unnumbered chapters and sections and with \phantomsection. The following key allows to force a specific name for the destination so that it can be used by bookmarks.

```
179 \keys_define:nn { hyp / setup }
180 {
181     next-anchor .code:n =
182     {
183         \AddToHookNext{__hyp/dest/make}
184         {\Hy@MakeCurrentHref{#1}}
185     }
186 }
```

Allow non-ascii in href, and add more href versions. We add a few new keys: `urlencode` to force percent encoding (`\hrefurl`, `\href`) `protocol` to add a protocol (`\hrefurl`, `\href` doesn't work here as it needs the colon for the split and the guessing.) `destination` to add a destination (`\hrefpdf`)

```

187
188 \bool_new:N \l__hyp_href_url_encode_bool
189 \bool_new:N \l__hyp_href_url_ismap_bool
190 \tl_new:N \l__hyp_href_url_protocol_tl
191 \tl_new:N \l__hyp_href_pdf_destination_tl
192 \tl_new:N \l__hyp_href_pdf_page_tl
193 \tl_new:N \l__hyp_href_run_parameter_tl
194 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
195
196
197 \keys_define:nn { hyp / href }
198 {
199   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
200   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
201   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
202   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
203   ,pdfremotestartview .code:n =
204     {
205       \keys_set:nn { hyp / setup }
206       { pdfremotestartview = #1 }
207     }
208   ,page .code:n =
209     {
210       \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
211       \tl_set:Nn \Hy@href@page {#1}
212     }
213   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
214   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
215   ,nextactionraw .code:n =
216     { %perhaps some safety match later, see hyperref code
217       \tl_if_empty:nTF {#1}
218         {
219           \pdfdict_remove:nn{l_hyp/annot/A}{Next}
220         }
221         {
222           \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
223           \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
224           \keys_set:nn { hyp / setup }{ pdfnewwindow = true}
225         }
226     }
227   ,afrelationship .code:n =
228     {
229       \pdfdict_put:nnx
230       { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
231     }
232 }
233 }
234
235 \keys_define:nn { hyp / setup }

```

```

236 {
237   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
238   ,href / urlencode .default:n = {true}
239   ,href / urlencode .initial:n = {false}
240   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
241   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
242   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
243 }
244
245 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
246
247 \DeclareRobustCommand*{\href}[1][ ]{%
248   \mode_leave_vertical:
249   \hook_use:n{cmd/href/before}
250   \group_begin:
251   \keys_set:nn { hyp / href } {#1}
252   \let\protect\@unexpandable@protect
253   \bool_if:NTF \l__hyp_href_url_encode_bool
254     {
255       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
256     }
257     {
258       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
259     }
260   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
261 }
262
263 \begingroup
264   \catcode'\$=6 %
265   \catcode'\#=12 %
266   \gdef\href@\$1{\expandafter\href@split$1##\}%
267   \gdef\href@split$1#$2#$3\\$4{%
268     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
269   \endgroup
270   \hook_use:n{cmd/href/after}
271 }%
272 \endgroup
273
274 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
275
276 \DeclareRobustCommand*{\hrefurl}[1][ ]
277 {
278   \mode_leave_vertical:
279   \hook_use:n{cmd/href/before}
280   \group_begin:
281   \keys_set:nn { hyp / href } {#1}
282   \let\protect\@unexpandable@protect
283   \bool_if:NTF \l__hyp_href_url_encode_bool
284     {
285       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
286     }
287     {
288       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
289     }

```

```

290 \hyper@normalise\__hyp_href_url_aux:nn}
291
292 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
293 {
294   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
295   \group_end:
296   \hook_use:n{cmd/href/after}
297 }
298
299 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
300 \DeclareRobustCommand*{\hrefpdf}[1] []
301 {
302   \mode_leave_vertical:
303   \hook_use:n{cmd/hrefpdf/before}
304   \group_begin:
305   \keys_set:nn { hyp / href } {#1}
306   \let\protect\@unexpandable@protect
307   \hyper@normalise\__hyp_href_pdf_aux:nn
308 }
309
310 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
311 {
312   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
313   \group_end:
314   \hook_use:n{cmd/hrefpdf/after}
315 }
316
317 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
318 \DeclareRobustCommand*{\hrefrun}[1] []
319 {
320   \mode_leave_vertical:
321   \hook_use:n{cmd/hrefrun/before}
322   \group_begin:
323   \keys_set:nn { hyp / href } {#1}
324   \let\protect\@unexpandable@protect
325   \hyper@normalise\__hyp_href_run_aux:nn
326 }
327
328 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
329 {
330   \exp_args:Nnno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
331   \group_end:
332   \hook_use:n{cmd/hrefrun/after}
333 }
334
335
336 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
337
338 \DeclareRobustCommand*{\url}[1] []
339 {
340   \mode_leave_vertical:
341   \hook_use:n{cmd/url/before}
342   \group_begin:
343   \keys_set:nn { hyp / href } {#1}

```

```

344 \let\protect \@unexpandable@protect
345 \bool_if:NTF \l__hyp_href_url_encode_bool
346 {
347   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
348 }
349 {
350   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
351 }
352 \hyper@normalise\__hyp_href_url_aux:n
353 }
354
355 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
356 {
357   \exp_args:Nno
358     \hyper@linkurl{\__hyp_href_url_format: {#1}}
359     {\l__hyp_href_url_protocol_tl#1}
360   \group_end:
361   \hook_use:n{cmd/url/after}
362 }
363

```

the `\urldef` command doesn't like the optional argument, so we overwrite locally the `\url` command here:

```

364
365 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
366   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
367   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
368   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{#1}}}%
369   #2}
370

```

make the new commands compatible with `\pdfstringdef`:

```

371 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
372 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
373 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
374 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

## 4 Compatibility commands

### 4.1 Metadata

A number of values should be accessible from other packages. Until now packages like `hyperxmp` used variables like `\@pdfauthor`. As they are gone we need to provide some other access.

```

375 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
376 {
377   %\tl_set:cn {#1}{#2}
378   \AddToDocumentProperties[hyperref]{#1}{#2}
379 }
380 \cs_generate_variant:Nn \__hyp_store_metadata:nn {xn,nx,xx}

```

## 4.2 citecolor

cite is a link context. So we define a hook, and the keys in terms of this hook.

```

381 \hook_new:n{hyp/link/cite}
382 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
383 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
384 \keys_define:nn { hyp / setup }
385 {
386   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init
387   ,citebordercolor
388   .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_i
389 }
390 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
391 {
392   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
393   {
394     \keys_set:nn { hyp / setup }
395     {
396       linkcolor      = hyp/color/cite
397     }
398   }
399   \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
400 }
401 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
402 {
403   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
404   {
405     \keys_set:nn { hyp / setup }
406     {
407       linkbordercolor = hyp/color/citeborder
408     }
409   }
410   \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
411 }
412

```

## 5 Checks

The driver can not work properly if the pdfmanagement is not active, as keys need to write to the catalog and to info. But annotations and outlines should work. So should this be a fatal error? Should there be a difference between missing and inactive management? TODO

```

413 \bool_lazy_and:nnF
414 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
415 { \msg_error:nn { hyp}{ missing-resource-management } }

```

Outlines/bookmarks require the bookmark package. TODO check pdfpagemode if bookmarks are suppressed. TODO We overwrite the color key here for now, but this should be moved to bookmark

```

416 \AddToHook { package / after / bookmark}
417 {
418   \define@key{BKM}{color}

```

```

419     {
420       \__hyp_color_set:ne {__hyp/tmpa}{#1}
421       \color_export:nVN
422         {__hyp/tmpa}
423         \g__hyp_bordercolormodel_str
424         \BKM@color
425     }
426 }
427 \legacy_if:nT { Hy@bookmarks }
428 {
429   \AddToHook{begindocument/before}[hyperref/bookmark]
430   {
431     \RequirePackage{bookmark}
432   }
433 }
434 \legacy_if:nT { Hy@draft }
435 {
436   \PassOptionsToPackage{draft}{bookmark}
437 }

```

## 6 Reference and label commands

The code uses the l3ref-tmp package which must have been loaded as pdfmanagement is a requirement. The commands use after the module prefix always `_ref`.

At first a label command which add the space commands from LaTeX:

```

\__hyp_ref_label:nn
\__hyp_ref_if_exist:nn 438 %
\__hyp_ref_check:nn    439 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn    440 {
441   \@bsphack
442   \ref_label:nn{#1}{#2}
443   \@esphack
444 }
445 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

This provides a condition which tests if a label/attribute combination is known

```

446 \prg_new_eq_conditional:Nmn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
447 \prg_generate_conditional_variant:Nmn \__hyp_ref_if_exist:nn {en} { p , T , F , TF }

```

This checks if the label/attribute is known and issues a warning if not. It then also triggers the standard rerun message.

```

448 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
449 {
450   \__hyp_ref_if_exist:nnF {#1}{#2}
451   {
452     \protect\G@refundefinedtrue
453     \@latex@warning
454     {
455       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
456       on~page~\thepage~\space undefined
457     }
458   }
459 }

```



```
460 \cs_generate_variant:Nn \_hyp_ref_check:nn {en}
```

This retrieves a value, it is a simple wrapper around the `\ref_value:nn`

```
461 \cs_new:Npn \_hyp_ref_value:nn #1 #2
462 {
463   \ref_value:nn{#1}{#2}
464 }
465 \cs_generate_variant:Nn \_hyp_ref_value:nn {en}
```

*(End definition for \\_hyp\_ref\_label:nn and others.)*

## 7 Variables

### 7.1 Private temporary variables

At first a few generic tmp variables

```
\l__hyp_tmpa_tl
\l__hyp_tmpa_seq 466 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 467 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 468 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 469 \int_new:N \l__hyp_tmpa_int
470 \str_new:N \l__hyp_tmpa_str
```

*(End definition for \l\_\_hyp\_tmpa\_tl and others.)*

A number of more specific tmp variables. These will perhaps disappear or change.

```
\l__hyp_dest_name_tmpa_tl  TODO: document and check use!
  \l__hyp_uri_tmpa_tl      471 \tl_new:N \l__hyp_dest_name_tmpa_tl
  \l__hyp_filename_tmpa_tl 472 \tl_new:N \l__hyp_uri_tmpa_tl
\_hyp_text_tmpa_str\g__hyp_text_tmpa_str 473 \tl_new:N \l__hyp_filename_tmpa_tl
474 \tl_new:N \l__hyp_para_tmpa_tl
475 \str_new:N \l__hyp_text_tmpa_str
476 \str_new:N \g__hyp_text_tmpa_str
```

*(End definition for \l\_\_hyp\_dest\_name\_tmpa\_tl and others.)*

### 7.2 Constants

`\c__hyp_dest_undefined_tl` This variable is used if a destination name is empty.

```
477 \tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}
```

*(End definition for \c\_\_hyp\_dest\_undefined\_tl.)*

`\c__hyp_annot_types_seq` This constants holds the link types managed by hyperref along with a mapping from annot names to hyperref names and back.

```
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop
478 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
479 {url,link,file,menu,run}
480 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
481 {
482   URI    = url,
483   GoTo   = link,
484   GoToR  = file,
485   Named  = menu,
486   Launch= run
```

```

487 }
488 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
489 {
490   url    = URI,
491   link   = GoTo,
492   file   = GoToR,
493   menu   = Named,
494   run    = Launch
495 }
496

```

(End definition for `\c__hyp_annot_types_seq`, `\c__hyp_map_annot_hyp_prop`, and `\c__hyp_map_hyp_annot_prop`.)

### 7.3 Variables

`\g__hyp_dest_pdfstartpage_tl` The first holds the (absolute) start page number, the other the startview instruction for the current and remote files. The instruction is in “PDF format” but without the leading slash!

```

\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_tl
497 \tl_new:N \g__hyp_dest_pdfstartpage_tl
498 \tl_new:N \g__hyp_dest_pdfstartview_tl
499 \tl_new:N \l__hyp_dest_pdfremotestartview_tl

```

(End definition for `\g__hyp_dest_pdfstartpage_tl`, `\g__hyp_dest_pdfstartview_tl`, and `\l__hyp_dest_pdfremotestartview_tl`.)

It is still unclear which str convert option is the best in the various places, so we use a variable to allow tests and perhaps external configuration. The “print” type should always have the delimiters.

```

\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
500 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
501 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
502 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
503 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
504 \tl_new:N \l__hyp_text_enc_file_print_tl
505 \tl_new:N \l__hyp_text_enc_para_print_tl
506
507 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
508 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
509 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
510 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
511 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
512 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

(End definition for `\l__hyp_text_enc_uri_print_tl` and others.)

`\l__hyp_dest_pdfview_tl` This hold the destination instructions in a format suitable for `\pdf_destination:nn`. The special value `fitrbox` indicates a boxed destination.

```

513 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End definition for `\l__hyp_dest_pdfview_tl`.)

`hyp/annot/link` (*color name*) These color names are used for the annotations (colorlinks). They are initialized at the end when the color scheme is used

`hyp/annot/url` (*color name*)

`hyp/annot/file` (*color name*)

`hyp/annot/run` (*color name*)

`hyp/annot/menu` (*color name*)

`\g__hyp_bordercolormodel_str` This holds the export model for border color etc. It is currently either `space-sep-cmyk` or `space-sep-rgb`. The default is the second. It can be change by the key `bordercolormodel`

```
514 \str_new:N \g__hyp_bordercolormodel_str
```

(End definition for `\g__hyp_bordercolormodel_str`.)

## 7.4 Booleans

`\l_hyp_annot_colorlink_bool` `\l_hyp_annot_colorurl_bool` `\l_hyp_annot_colorfile_bool` `\l_hyp_annot_colorrund_bool` `\l_hyp_annot_colormenu_bool` These booleans are needed to control the colors. They are public so that other packages can query the state too.

```
515 \seq_map_inline:Nn \c__hyp_annot_types_seq
516 {
517   \bool_new:c {l_hyp_annot_color#1_bool}
518 }
```

(End definition for `\l_hyp_annot_colorlink_bool` and others. These variables are documented on page ??.)

`\l_hyp_annot_ocgcolorlink_bool` `\l_hyp_annot_ocgcolorurl_bool` `\l_hyp_annot_ocgcolorfile_bool` `\l_hyp_annot_ocgcolorrund_bool` `\l_hyp_annot_ocgcolormenu_bool` These booleans are needed to control the ocgcolors. They are public so that other packages can query the state too.

```
519 \seq_map_inline:Nn \c__hyp_annot_types_seq
520 {
521   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
522 }
```

(End definition for `\l_hyp_annot_ocgcolorlink_bool` and others. These variables are documented on page ??.)

`\not_Named_bool_{...}` `\l_hyp_annot_Launch_bool` This booleans are used to disable some link types while keeping others.

```
523 \seq_map_inline:Nn \c_pdfannot_link_types_seq
524 {
525   \bool_new:c {l__hyp_annot_#1_bool}
526   \bool_set_true:c {l__hyp_annot_#1_bool}
527 }
```

(End definition for `\l__hyp_annot_GoTo_bool` `\l__hyp_annot_URI_bool` `\l__hyp_annot_GoToR_bool` `\l__hyp_annot_Named_bool` `\l__hyp_annot_Launch_bool`.)

## 7.5 Boxes

`\l__hyp_dest_box` This holds an (empty) box which is used to get the width for FitR destinations.

```
528 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

## 7.6 Regex

`\c__hyp_dest_startview_regex` This regex is used to extract the right arguments `pdfstartview` and `pdfremotestartview`. Their values is filled up with `null` and then the start extracted.

```
529 \regex_const:Nn \c__hyp_dest_startview_regex
530 {
531   \A\ *
532   (?
533     (?:XYZ (?:\ +(?:?:\d+|\d*\.\d+)|null)){3}\ )
534     |
535     (?:Fit\b|FitB\b)
536     |
537     (?:?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
538     |
539     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
540   )
541 }
```

*(End definition for `\c__hyp_dest_startview_regex`.)*

## 7.7 PDF dictionaries

`l__hyp_page/Trans` This dictionary is used for page transitions.

```
542 \pdfdict_new:n {l__hyp_page/Trans}
543 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

*(End definition for `l__hyp_page/Trans`.)*

## 8 PDF string conversion

This defines a command which is used to replace `\pdfstringdef`. This is probably temporary and will be adjusted or replaced if some more generic PDF string command/module exists. All commands here use the “submodule” name `text`. At first a hook for user additions:

`hyp/text/pdfstring`

```
544 \hook_new:n {hyp/text/pdfstring}
```

*(End definition for `hyp/text/pdfstring`. This function is documented on page ??.)*

The first step to convert input in a PDF string is to purify it, that means to remove/expand commands. As the whole process is not expandable anyway we can use a protected command. The “output” is a string:

`\__hyp_text_purify:nN`

```
545 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
546 {
547   \str_set:Nx #2 {\text_purify:n { #1 } }
548 }
```

*(End definition for `\__hyp_text_purify:nN`.)*

The second step is to cleanup the output of the first step. This is a dummy currently. The argument should be a string variable.

`\__hyp_text_cleanup:N`

```
549 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
550 {
551
552 }
```

*(End definition for \\_\_hyp\_text\_cleanup:N.)*

The last step converts the string to a PDF encoding. As we have at least two targets (hex and literal) there is an argument. The conversion assumes utf8 input, it is based on `cspdf_string_from_unicode:nnN` in `l3pdfutils`.

#2 is str variable, #1 should be one of

<code>utf8/string</code>	<code>(lit)</code> <code>(utf8/string)</code>
<code>utf8/string-raw</code>	<code>lit</code> <code>(utf8/string)</code>
<code>utf8/URI</code>	<code>(percent encoded url)</code>
<code>utf8/URI-raw</code>	<code>percent encoded url</code>
<code>utf16/hex</code>	<code>&lt;HEX&gt;</code> <code>(utf16/hex)</code>
<code>utf16/hex-raw</code>	<code>HEX</code> <code>(utf16/hex)</code>
<code>utf16/string</code>	<code>(lit)</code> <code>(utf16/string)</code>
<code>utf16/string-raw</code>	<code>lit</code> <code>(utf16/string)</code>

`\__hyp_text_string_from_unicode:nN`

```
553 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
554 {
555   \pdf_string_from_unicode:nVN { #1 } #2 #2
556 }
```

*(End definition for \\_\_hyp\_text\_string\_from\_unicode:nN.)*

This command combines everything. #1=input, #2= handler shortcut #3= output str variable The command uses a group to locally set `\Hy@pdfstringtrue` so that `\texorpdfstring` works and other local settings can be done.

`\__hyp_text_pdfstring:nnN`

```
557 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
558 {
559   \group_begin:
560   \Hy@pdfstringtrue
561   \hook_use:n {hyp/text/pdfstring}
562   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
563   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
564   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
565   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
566   \group_end:
567   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
568 }
569 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {xnN,onN,xoN,ooN,noN}
```

*(End definition for \\_\_hyp\_text\_pdfstring:nnN.)*

!!! temporary until all instances are gone

```
570 \cs_new_protected:Npn \Hy@pstringdef #1 #2
571 { \__hyp_text_pdfstring:xnN {#2} {utf8/string-raw}#1 }
```

This is a special version for info keys:

```
\__hyp_text_pdfstring_info:nN
```

```
572 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
573 {
574   \__hyp_text_pdfstring:noN { #1 } { \l__hyp_text_enc_info_print_tl } #2
575 }
```

(End definition for `\__hyp_text_pdfstring_info:nN`.)

## 9 Pagelabels

Page labels are representations of the page numbers in the PDF viewer. If the hyperref options `pdfpagelabels` is true (the default) roman numbers are e.g. shown as “ii (2/58)”. To do this the page ranges must be collected, if possible a prefix and the numbering of the counter must be identified and then written to the catalog.

The current implementation in hyperref/hyperref drivers:

**xetex:** hxdetex.def, line 80-110

`\HyPL@StorePageLabel` writes to the aux-file at begin document (after reading the aux) `\HyPL@SetPageLabels` is called (defined in hyperref.sty after the driver loading) which calls `\Hy@PutCatalog{/PageLabels<</Nums[\HyPL@Labels]>>}`

**dvips:** identical to xetex, line 60 to 90 in pdfmark.def

**dvipdfm:** identical to xetex

**pdftex:** `\HyPL@StorePageLabel` stores in `\HyPL@Labels` in the first compilation In `\AtVeryEndDocument` `\HyPL@SetPageLabels` is called.

**luatex** identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-30 is correctly split. If the counter can not be identified hyperref generates only /P entries with the whole content.

The new implementation makes use of the pdf management: The relevant entry in the catalog is continuously updated and pushed out at the end of the document. This works (hopefully ...) with all drivers.

We do not try to avoid the (in hyperref’s wording) “useless” pagelabel entry `/PageLabels <</Nums[0<</S/D>>]>>` (but it would be possible), we also don’t test for empty `\thepage`, hyperref seems to handle this fine and the pdf is valid.

The code has to define `\Hy@PutCatalog` as we can’t yet change code in hyperref. The switch for draftmode has been removed.

```
\__hyp_PageLabels_gpush:
```

```
\Hy@PutCatalog
\HyPL@StorePageLabel
576 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
577 {
578   \pdfmanagement_add:nx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
579 }
580
581 \def\Hy@PutCatalog #1 {}
582
583
584 \legacy_if:nT { Hy@pdfpagelabels }
585 {
```

```

586 \cs_set_protected:Npn \HyPL@StorePageLabel #1
587 {
588   \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
589   \__hyp_PageLabels_gpush:
590 }
591 }

```

(End definition for `\__hyp_PageLabels_gpush:`, `\Hy@PutCatalog`, and `\HyPL@StorePageLabel`. These functions are documented on page ??.)

## 10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

\hyper@anchor
\hyper@anchorstart
\hyper@anchorend
\hyper@link      %GoTo
\hyper@linkstart %GoTo
\hyper@linkend   %GoTo
\hyper@linkfile  %GoToR
\hyper@linkurl   %URI

```

This driver defines for consistency also `\hyper@linklaunch` for Launch and `\hyper@linknamed` for Named.

### 10.1 Anchors / destinations

The first three commands are needed for “anchors”. At first the internal commands to create a destination. It uses `\Hy@WrapperDef` to make it babel safe, it is not clear if this is still needed, but we leave it for now.

---

```

\__hyp_destination:nn \__hyp_destination:nn {<destination name>} {<location>}

```

The `<destination name>` is encoded with the method stored in `\l__hyp_text_enc_dest_tl`. The location should be one of `fit`, `fith`, `fitv`, `fitbv`, `fitbh`, `fitr`, `xyz`, `fitrbx`. The last will make use of `\l__hyp_dest_box`

```

\__hyp_destination:nn

```

```

592 \Hy@WrapperDef \__hyp_destination:nn #1 #2
593 {
594   \mode_if_horizontal:T { \@savs\spacefactor }
595   \Hy@SaveLastskip      %defined in hyperref
596   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
597   \__hyp_text_pdfstring:xoN
598   { \HyperDestNameFilter{#1} }
599   { \l__hyp_text_enc_dest_tl }
600   \l__hyp_tmpa_tl
601   \str_if_eq:nnTF {#2} {fitrbox}
602   {
603     \exp_args:NV
604     \pdf_destination:nnnn \l__hyp_tmpa_tl
605     { \box_wd:N \l__hyp_dest_box }

```

```

606     { \box_ht:N \l__hyp_dest_box }
607     { \box_dp:N \l__hyp_dest_box }
608   }
609   {
610     \exp_args:NV
611     \pdf_destination:nf
612     { \l__hyp_tmpa_tl }
613     { #2 }
614   }
615   \Hy@RestoreLastskip   %defined in hyperref
616   \mode_if_horizontal:T { \spacefactor\@savs f }
617 }

```

(End definition for `\__hyp_destination:nn`.)

This are the three destinations commands. They are modelled along the xetex version. It is not quite clear if really all three are needed for the backends supported by this driver, but changing the hyperref code would be difficult. We add a hook. This allows e.g. the tagging code to create also a structured destination. We don't use the cmd hook, as we want the same hook for both start commands. We make the current dest name available so that the hook code can use it.

```

\hyper@anchor
\hyper@anchorstart 618 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 619 \hook_new:n{hyp/anchor}
hyp/anchor 620 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 621 {
622   \exp_args:NnV
623   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
624   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
625   \hook_use:n{hyp/anchor}
626 }
627
628 \cs_new_protected:Npn \hyper@anchorstart #1
629 {
630   \Hy@activeanchortrue
631   \exp_args:NnV
632   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
633   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
634   \hook_use:n{hyp/anchor}
635 }
636
637 \cs_new_protected:Npn \hyper@anchorend
638 {
639   \Hy@activeanchorfalse
640 }

```

(End definition for `\hyper@anchor` and others. These functions are documented on page ??.)

## 10.2 GoTo Links

The next three commands are for links inside the document, to destinations (GoTo links). The definition in `hyperref` have a first argument which can be used to pass a semantical context. Currently this argument is only used for `\cite` and only to change the color. The new implementation uses it for a real hook.



At first the internal link commands:

```

641 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
642 {
643   \mode_leave_vertical:
644   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
645   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
646   {
647     \msg_warning:nxx
648     { hyp }
649     { empty-destination-name }
650     { \c__hyp_dest_undefined_tl }
651     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
652   }
653   {
654     \__hyp_text_pdfstring:xoN
655     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
656     { \l__hyp_text_enc_dest_tl }
657     \l__hyp_dest_name_tmpa_tl
658   }
659   \exp_args:No
660   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
661 }
662
663 \cs_new_protected:Npn \__hyp_link_goto_end:
664 {
665   \pdfannot_link_goto_end:
666 }

```

Now the three hyperref commands. The splitted commands `\hyper@linkstart` and `\hyper@linkend` are used for footnotemarks, toc and natbib-cites.

---

`\hyper@link` `\hyper@link{<context>}{<destination name>}{<link text>}`

This creates a complete GoTo link around the `<link text>` pointing to `<destination name>`. The hook `hyp/link/<context>` is executed at the begin if it exists.

The only `<context>` for which a hook is predefined is `cite`. Packages which want to use another `<context>` should initialize the hook like this:

```

\IfHookExistsTF{hyp/link/context}{ }
{ \NewHook{hyp/link/context} }

```

The hook code is executed in a group but before all the pdfannot hooks.

---

`\hyper@linkstart` `\hyper@linkstart{<context>}{<destination name>}`  
`\hyper@linkend` `\hyper@linkend`

This creates the start and end commands for a GoTo link around the text between both pointing to `<destination name>`. The hook `hyp/link/<context>` is executed at the begin if it exists as with `\hyper@link`

The commands open and close a group, so should be placed carefully. .

hyperref adds a group with `\Hy@colorlink`, we move this outside the link so that it groups the context hook too. We store again the destination name in the public `tl \l_hyp_current_dest_name_tl` so that the hook code can make use of it

667

```

668 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
669 {
670   \bool_if:NTF \l__hyp_annot_GoTo_bool
671   {
672     \Hy@VerboseLinkStart{#1}{#2}
673     \group_begin:
674     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
675     \hook_use:n {hyp/link/#1}
676     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
677     \__hyp_link_goto_end:
678     \group_end:
679     \Hy@VerboseLinkStop
680   }
681   {\let\protect\relax#3}
682 }
683
684 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
685 {
686   \bool_if:NT \l__hyp_annot_GoTo_bool
687   {
688     \Hy@VerboseLinkStart{#1}{#2}% only for debug
689     \group_begin:
690     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
691     \hook_use:n {hyp/link/#1}
692     \__hyp_link_goto_begin:nw {#2}
693   }
694 }
695
696 \cs_new_protected:Npn \hyper@linkend
697 {
698   \bool_if:NT \l__hyp_annot_GoTo_bool
699   {
700     \__hyp_link_goto_end:
701     \group_end:
702     \Hy@VerboseLinkStop
703   }
704 }

```

### 10.3 URI links

We define a dictionary for the action dictionary. For now it is public.

```

705 \pdfdict_new:n {l_hyp/annot/A/URI}
706 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
707 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
708
709 \cs_new_protected:Npn \hyper@linkurl #1 #2 % #1: link text #2: URI,
710 {
711   \bool_if:NTF \l__hyp_annot_URI_bool
712   {
713     \group_begin:
714     \__hyp_text_pdfstring:xoN
715     { #2}
716     { \l_hyp_text_enc_uri_print_tl }
717     \l_hyp_uri_tmpa_tl

```

```

718 \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
719 \bool_if:NT \l__hyp_href_url_ismap_bool
720 {
721   \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
722 }
723 \cs_set_eq:NN \# \c_hash_str
724 \cs_set_eq:NN \% \c_percent_str
725 \Hy@safe@activetrue
726 \mode_leave_vertical:
727 \pdfannot_dict_put:nmx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
728 \pdfannot_link:nxn { URI }
729 {
730 }
731 {
732   \let\protect\relax
733   #1
734   \Hy@xspace@end
735   \Hy@VerboseLinkStop %where is the start??
736 }
737 \group_end:
738 }
739 {\let\protect\relax#1}}
740 }
741

```

## 10.4 GoToR Links files

```

742 \pdfdict_new:n {l_hyp/annot/A/GoToR}
743 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
744 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
745
746
747 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
748 {
749   \bool_if:NTF \l__hyp_annot_GoToR_bool
750   {
751     \group_begin:
752     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
753     \exp_args:Nx
754     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
755     {
756       \pdfdict_put:nmx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap}}
757       \exp_args:Nnox
758       \pdffile_embed_file:nnn
759       {}
760       {\l__hyp_filename_tmpa_tl }
761       {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
762     }
763     \pdfdict_put:nmx
764     {l_hyp/annot/A/GoToR}
765     {F}
766     {\pdf_object_ref:e {\__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
767     \__hyp_text_pdfstring:nnN
768     { #3 }

```

```

769     { \l__hyp_text_enc_dest_print_t1 }
770     \l__hyp_dest_name_tmpa_t1
771 \tl_if_blank:eTF {#3}
772 {
773     \pdfdict_put:nxx {l_hyp/annot/A/GoToR}{D}
774     {
775         [
776             \int_eval:n
777             { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_t1 - 1 }}
778             /\l__hyp_dest_pdfremotestartview_t1
779         ]
780     }
781 }
782 {
783     \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_t1}
784 }
785 \mode_leave_vertical:

```

We use an extra object here, as ghostscript doesn't like the object reference in the dict <https://chat.stackexchange.com/transcript/message/57361080#57361080>

```

786     \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
787     \pdfannot_dict_put:nxx {link/GoToR}{A}{\pdf_object_ref_last:}
788     \pdfannot_link:nxn %expansion??
789     { GoToR }
790     {
791     }
792     {
793         \let\protect\relax
794         #1\Hy@xspace@end
795         \Hy@VerboseLinkStop %where is the start??
796     }
797     \group_end:
798 }
799 {{\let\protect\relax#1}}
800 }

```

## 10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

801 \pdfdict_new:n {l_hyp/annot/A/Launch}
802 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
803 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
804
805 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
806 {
807     \bool_if:NTF \l__hyp_annot_Launch_bool
808     {
809         \group_begin:
810         \__hyp_text_pdfstring:nnN
811         { #1 }
812         { \l__hyp_text_enc_file_print_t1 }
813         \l__hyp_filename_tmpa_t1
814         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_t1}
815         \__hyp_text_pdfstring:noN
816         { #3 }

```

```

817         { \l__hyp_text_enc_para_print_tl }
818         \l__hyp_para_tmpa_tl
819     \bool_if:nTF
820     {
821         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
822         ||
823         \pdf_version_compare_p:Nn > {1.9}
824     }
825     {
826         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
827     }
828     {
829         \pdfdict_put:nxx {l_hyp/annot/A/Launch}{Win}
830         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
831     }
832     \mode_leave_vertical:
833     \pdfannot_dict_put:nxx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
834     \pdfannot_link:nxn
835     { Launch }
836     {
837     %         /A
838     %         <<
839     %         \pdfdict_use:n {l_hyp/annot/A/Launch}
840     %         >>
841     }
842     {
843         \let\protect\relax
844         #2\Hy@xspace@end
845         \Hy@VerboseLinkStop %where is the start??
846     }
847     \group_end:
848     }
849     {\let\protect\relax#2}}
850 }

```

The actually command used by hyperref is `\@hyper@launch` which uses a delimited argument, because of the color the definition is a bit convoluted.

```

851 \use:x
852 { % filename, anchor text, linkname
853     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
854 }
855 {
856     \hyper@linklaunch {#1}{#2}{#3}
857 }

```

## 10.6 Named links (menu)

We also define `\hyper@linknamed` for consistency.

```

858 \pdfdict_new:n {l_hyp/annot/A/Named}
859 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
860 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
861
862 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
863 {

```

```

864 \bool_if:NTF \l__hyp_annot_Named_bool
865 {
866   \group_begin:
867   \pdfmeta_standard_verify:nnTF {named_actions}{#1}
868   {
869     \mode_leave_vertical:
870     \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
871     {\pdf_name_from_unicode_e:n{#1}}
872     \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
873     \pdfannot_link:nxn { Named }
874     {
875       % /A
876       % <<
877       % \pdfdict_use:n { l_hyp/annot/A/Named }
878       % >>
879     }
880     {
881       #2
882       \Hy@xspace@end
883       \Hy@VerboseLinkStop
884     }
885   }
886   {
887     \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
888     #2
889   }
890   \group_end:
891 }
892 {\let\protect\relax#2}}
893 }
894

```

## 11 Link decorations

### 11.1 Functions to export and select colors

We support two input syntax: color expressions and model with values. Exporting can be done by first setting the color with `\__hyp_color_set:nn` (if needed to a temporary color name) and then using `\color_export:nnN`. But we need a variant as the export format `space-sep-cmyk` or `space-sep-rgb` is stored in a tl.

```

895 \cs_generate_variant:Nn \color_export:nnN {nVN}

```

---

```

\__hyp_color_select:n \__hyp_color_select:n {<color>}

```

These commands select a (text) color. `{<color>}` should have either the format `[model]{value}` or be a color expression. For examples: `[rgb]{1,0,.5}` or `red!50!blue`

```

\__hyp_color_select:n Color keys need to parse color expressions. Two input types are supported: color=[rgb]{1,0,.5}
\__hyp_color_select_aux:wn and color=red!50!blue.

```

```

896 \cs_new_protected:Npn \__hyp_color_select:n #1
897 {
898   \tl_if_head_eq_charcode:nNTF {#1}[ %]

```

```

899     {
900     \__hyp_color_select_aux:wn #1
901     }
902     {
903     \color_select:n {#1}
904     }
905   }
906
907 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
908   {
909     \color_select:nn {#1}{#2}
910   }
911
912 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

(End definition for `\__hyp_color_select:n` and `\__hyp_color_select_aux:wn`.)

---

`\__hyp_color_set:nn` `\__hyp_color_set:nn {< name >} {< color >}`

These commands store the color in `{< name >}`. `{< color >}` should have either the format `[model]{value}` or be a color expression. For examples: `[rgb]{1,0,.5}` or `red!50!blue`

`\__hyp_color_set:nn` Color keys need to parse color expressions. Two input types are supported: `color=[rgb]{1,0,.5}`  
`\__hyp_color_set_aux:nwn` and `color=red!50!blue`.

```

913 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
914   {
915     \tl_if_head_eq_charcode:nNTF {#2}[ %]
916     {
917       \__hyp_color_set_aux:nwn { #1 } #2
918     }
919     {
920       \color_set:nn {#1} {#2}
921     }
922   }
923
924 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
925   {
926     \color_set:nnn {#1}{#2}{#3}
927   }
928
929 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End definition for `\__hyp_color_set:nn` and `\__hyp_color_set_aux:nwn`.)

## 11.2 Textcolor of links

colors are added in the hooks. This means that they can also be removed if needed. They add a group—this isn't needed with `hyperref` code, but could be relevant with low-level annotations.

```

930 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
931   {
932     \hook_gput_code:nnn
933     {pdfannot/link/#2/begin}
934     {hyp/color}

```

```

935     {
936       \bool_if:cT { l_hyp_annot_color#1_bool }
937       {
938         \group_begin:
939         \color_select:n { hyp/color/#1}
940       }
941     }
942   \hook_gput_code:nnn
943   {pdfannot/link/#2/end}
944   {hyp/color}
945   {
946     \bool_if:cT { l_hyp_annot_color#1_bool }
947     {
948       \group_end:
949     }
950   }
951 }

```

`colorlinks` (*setup key*) This key also resets the border and borderstyle.

```

952 \keys_define:nn { hyp / setup }
953 {
954   ,colorlinks .meta:n =
955   {
956     ,pdfborder={0~0~0}
957     ,pdfborderstyle=
958     ,colorurl =#1
959     ,colorlink =#1
960     ,colorrurl =#1
961     ,colormenu =#1
962     ,colorfile =#1
963   }
964   ,colorlinks .default:n = {true}
965 }

```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorrurl` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolors` (*setup key*)

```

966 \seq_map_inline:Nn \c__hyp_annot_types_seq
967 {
968   \keys_define:nn { hyp / setup }
969   {
970     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
971     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
972   }
973 }
974
975 \keys_define:nn { hyp / setup }
976 {
977   ,allcolors .meta:n =
978   {
979     ,urlcolor=#1
980     ,linkcolor=#1
981     ,runcolor=#1
982     ,filecolor=#1
983     ,menucolor=#1
984   }

```



```

985     ,allcolors .value_required:n = true
986   }

```

## 11.3 Style and color of borders

### 11.3.1 Border color

The border color is set by link type. The color can be set as rgb (default) or cmyk (unusual). This can be set with the `bordercolormodel` key:

`bordercolormodel` (*setup key*)

```

987 \keys_define:nn { hyp / setup }
988   {
989     ,bordercolormodel .choices:nn =
990     {rgb,cmyk}
991     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
992     ,bordercolormodel .initial:n ={rgb}
993   }

994 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
995   {
996     \keys_define:nn { hyp / setup }
997     {
998       #1bordercolor .code:n =
999       {
1000         \tl_if_empty:nTF { ##1 }
1001         {
1002           \pdfannot_dict_remove:nn
1003           {link/#2}
1004           { C }
1005         }
1006         {
1007           \__hyp_color_set:ne {hyp/color/#1border}{##1}
1008           \color_export:nVN
1009           {hyp/color/#1border}
1010           \g__hyp_bordercolormodel_str
1011           \l__hyp_tmpa_tl
1012           \pdfannot_dict_put:nnx
1013           {link/#2}
1014           { C }
1015           { [\l__hyp_tmpa_tl] }
1016         }
1017       }
1018     }
1019   }
1020
1021 \keys_define:nn { hyp / setup }
1022   {
1023     ,allbordercolors .meta:n =
1024     {
1025       ,linkbordercolor=#1
1026       ,urlbordercolor =#1
1027       ,filebordercolor=#1
1028       ,menubordercolor=#1
1029       ,runbordercolor =#1

```

```

1030     }
1031     ,allbordercolors .value_required:n = true
1032   }
1033

```

### 11.3.2 Borderwidth and -arc

```

1034 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1035 {
1036   \keys_define:nn { hyp / setup }
1037   {
1038     #1border .code:n =
1039     {
1040       \tl_if_empty:nTF { ##1 }
1041       {
1042         \pdfannot_dict_remove:nn
1043         {link/#2}
1044         { Border }
1045       }
1046       {
1047         \pdfannot_dict_put:nnn
1048         {link/#2}
1049         { Border }
1050         { [##1] }
1051       }
1052     }
1053   }
1054 }
1055 \keys_define:nn { hyp / setup }
1056 {
1057   ,pdfborder .code:n =
1058   {
1059     \tl_if_empty:nTF { #1 }
1060     {
1061       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1062       {
1063         \pdfannot_dict_remove:nn
1064         {link/##2}
1065         { Border }
1066       }
1067     }
1068     {
1069       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1070       {
1071         \pdfannot_dict_put:nnn
1072         {link/##2}
1073         { Border }
1074         { [#1] }
1075       }
1076     }
1077   }
1078   ,pdfborder .initial:n = {0~0~1},
1079 }

```

### 11.3.3 Borderstyle

This keys fill the extended /BS entry (a dictionary).

```
pdfborderstyle (setup key)
urlborderstyle (setup key) 1080 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1081 {
runborderstyle (setup key) 1082   \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1083   {
menuborderstyle (setup key) 1084     #1borderstyle .code:n =
1085     {
1086       \tl_if_empty:nTF { ##1 }
1087       {
1088         \pdfannot_dict_remove:nn
1089         {link/#2}
1090         { BS }
1091       }
1092       {
1093         \pdfannot_dict_put:nnn
1094         {link/#2}
1095         { BS }
1096         { <<##1>> }
1097       }
1098     }
1099   }
1100 }
1101 \keys_define:nn { hyp / setup }
1102 {
1103   ,pdfborderstyle .code:n =
1104   {
1105     \tl_if_empty:nTF { #1 }
1106     {
1107       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1108       {
1109         \pdfannot_dict_remove:nn
1110         {link/##2}
1111         { BS }
1112       }
1113     }
1114     {
1115       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1116       {
1117         \pdfannot_dict_put:nnn
1118         {link/##2}
1119         { BS }
1120         { <<##1>> }
1121       }
1122     }
1123   }
1124   ,pdfborderstyle .initial:n = {},
1125 }
```

## 11.4 ocolorlinks

OCG colorlinks need objects and an entry in the catalog. Perhaps the objects need public names to avoid that ocgx2 has to create duplicates? TODO

`\__hyp_ocg_init:` This commands write the objects as needed if ocg links are used. The initialization should happens only once.

```
1126 \cs_new_protected:Npn \__hyp_ocg_init:
1127 {
1128   \pdf_object_new:nn { __hyp/OCG/View } { dict }
1129   \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1130   \pdf_object_new:nn { __hyp/OCG/config } { dict }
1131   \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1132   \pdf_object_write:nx { __hyp/OCG/refarray }
1133   {
1134     \pdf_object_ref:n { __hyp/OCG/View }
1135     \c_space_tl
1136     \pdf_object_ref:n { __hyp/OCG/Print }
1137   }
1138   \pdf_object_write:nn { __hyp/OCG/View }
1139   {
1140     /Type/OCG
1141     /Name(View)
1142     /Usage
1143     <<
1144       /Print <</PrintState/OFF>>~
1145       /View <</ViewState/ON >>~
1146     >>
1147   }
1148   \pdf_object_write:nn { __hyp/OCG/Print }
1149   {
1150     /Type/OCG
1151     /Name(Print)
1152     /Usage
1153     <<
1154       /Print <</PrintState/ON>>~
1155       /View <</ViewState/OFF>>~
1156     >>
1157   }
1158   \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View } }
1159   \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print } }
1160   \pdf_object_write:nx { __hyp/OCG/config }
1161   {
1162     /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1163     /AS[
1164       <<
1165         /Event/View
1166         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1167         /Category[/View]
1168       >>
1169       <<
1170         /Event/Print
1171         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1172         /Category[/Print]
```

```

1173         >>
1174         <<
1175         /Event/Export
1176         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1177         /Category[/Print]
1178         >>
1179         ]
1180     }
1181     \pdfmanagement_add:nnx { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1182     \cs_gset:Npn \__hyp_ocg_init: {}
1183 }

```

(End definition for \\_\_hyp\_ocg\_init:.)

We use like with colors a hook, this allows ocgx to replace it. The implementation is rather simple and uses a box.

```

1184 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1185 {
1186     \hook_gput_code:nnn
1187     {pdfannot/link/#2/begin}
1188     {hyp/ocg}
1189     {
1190         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1191         {
1192             \__hyp_ocg_init:
1193             \group_begin:
1194             \hbox_set:Nw \l__hyp_tmpa_box
1195         }
1196     }
1197     \hook_gput_code:nnn
1198     {pdfannot/link/#2/end}
1199     {hyp/ocg}
1200     {
1201         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1202         {
1203             \hbox_set_end:
1204             \mbox
1205             {
1206                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1207                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1208                 \pdf_emc:
1209                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1210                 \group_begin:
1211                 \color_select:n { hyp/color/#1 }
1212                 \box_use_drop:N \l__hyp_tmpa_box
1213                 \group_end:
1214                 \pdf_emc:
1215             }
1216             \group_end:
1217         }
1218     }
1219 }

```

ocgcolorlinks (setup key) These are the keys for ocgcolors. We try to disable it for pdf version below 1.5

ocgcolorlink (setup key) 1220 \bool\_lazy\_or:nnTF

ocgcolorurl (setup key)

ocgcolorfile (setup key)

ocgcolormenu (setup key)

ocgcolorrund (setup key)

```

1221 { \pdf_version_compare_p:Nn > {1.4} }
1222 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1223 {
1224   \keys_define:nn { hyp / setup }
1225   {
1226     ,_ocgcolorlinks .meta:n =
1227     {
1228       ocgcolorlink=#1,
1229       ocgcolorurl=#1,
1230       ocgcolorfile=#1,
1231       ocgcolorrun=#1,
1232       ocgcolormenu=#1
1233     }
1234     ,_ocgcolorlinks .default:n = true
1235   }
1236 }
1237 {
1238   \keys_define:nn { hyp / setup }
1239   {
1240     ,_ocgcolorlinks .code:n =
1241     {
1242       \msg_warning:nnxx
1243       { hyp }
1244       { ignore-deprecated-or-unknown-option-in-pdf-version }
1245       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1246     }
1247   }
1248 }
1249
1250 \keys_define:nn { hyp / setup }
1251 {
1252   ,ocgcolorlinks .choice:
1253   ,ocgcolorlinks / true .meta:n =
1254   {
1255     pdfborder      ={{0~0~0}},
1256     pdfborderstyle ={{}},
1257     colorlinks     = false,
1258     _ocgcolorlinks = true
1259   }
1260   ,ocgcolorlinks / false .meta:n =
1261   {
1262     _ocgcolorlinks = false
1263   }
1264   ,ocgcolorlinks .default:n = {true}
1265 }
1266
1267 \seq_map_inline:Nn \c__hyp_annot_types_seq
1268 {
1269   \bool_lazy_or:nnTF
1270   { \pdf_version_compare_p:Nn > {1.4} }
1271   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1272   {
1273     \keys_define:nn { hyp / setup }
1274     {

```

```

1275         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1276     }
1277 }
1278 {
1279     \keys_define:nn { hyp / setup }
1280     {
1281         ,ocgcolor#1 .code:n=
1282         {
1283             \msg_warning:nxxx
1284             { hyp }
1285             { ignore-deprecated-or-unknown-option-in-pdf-version }
1286             { ocgcolor#1 }
1287             { \pdf_version_major:.\pdf_version_minor: }
1288         }
1289     }
1290 }
1291 }

```

## 11.5 Highlighting

This keys set what happens if you click on a link

```

1292 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1293 {
1294     \keys_define:nn { hyp / setup }
1295     {
1296         ,#1highlight .choices:nn =
1297         { /I, /N, /O, /P}
1298         {
1299             \pdfannot_dict_put:nnn
1300             {link/#2}
1301             { H }
1302             { ##1 }
1303         }
1304     }
1305     ,#1highlight / .code:n =
1306     {
1307         \pdfannot_dict_remove:nn
1308         {link/#2}
1309         { H }
1310     }
1311 }
1312     ,#1highlight / unknown .code:n =
1313     {
1314         \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1315         { pdfhighlight }
1316         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1317         { \exp_not:n {##1} }
1318     }
1319 }
1320 }
1321 }
1322 }
1323 \keys_define:nn { hyp / setup }
1324 {

```

```

1325 ,pdfhighlight .choices:nn =
1326 { /I, /N, /O, /P}
1327 {
1328   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1329   {
1330     \pdfannot_dict_put:nnn
1331     {link/###2}
1332     { H }
1333     { #1 }
1334   }
1335 }
1336 ,pdfhighlight / .code:n =
1337 {
1338   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1339   {
1340     \pdfannot_dict_remove:nn
1341     {link/##2}
1342     { H }
1343   }
1344 }
1345 ,pdfhighlight .initial:n = {/I},
1346 ,pdfhighlight / unknown .code:n =
1347 {
1348   \msg_warning:nxxx { hyp } { unknown-choice+empty }
1349   { pdfhighlight }
1350   { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1351   { \exp_not:n {#1} }
1352 }
1353 }

```

## 11.6 Hiding links

This key disable all appearance keys. The link themselves are still there.

```

hidelinks (setup key)
hidelink (setup key) 1354 \keys_define:nn { hyp / setup }
hideurl (setup key) 1355 {
hidefile (setup key) 1356   hidelinks .meta:n =
hiderun (setup key) 1357   {
hidemenu (setup key) 1358     ,colorlinks      = false
1359     ,ocgcolorlinks = false
1360     ,pdfborder      = { 0~0~0 }
1361     ,pdfborderstyle=
1362   }
1363 }
1364
1365 \seq_map_inline:Nn \c__hyp_annot_types_seq
1366 {
1367   \keys_define:nn { hyp / setup }
1368   {
1369     hide#1 .meta:n =
1370     {
1371       ,color#1      = false
1372       ,ocgcolor#1  = false

```



```

1373         ,#1border      = { 0~0~0 }
1374         ,#1borderstyle =
1375     }
1376 }
1377 }

```

## 11.7 color schemes and settings

This define the key for the color schemes and sets the default colors.

`colorscheme` (*setup key*)

```

1378 \keys_define:nn { hyp / setup }
1379 {
1380     colorscheme .code:n =
1381     {
1382         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1383         {
1384             \keys_set:nn { hyp /setup }
1385             {
1386                 ##1 = ##2
1387             }
1388         }
1389     }
1390 }
1391 \keys_set:nn { hyp / setup } {colorscheme=phelype}

```

## 12 Keys

### 12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode` (*setup key*)

`pdfencoding` (*setup key*)

`pdfversion` (*setup key*)

```

1392 \keys_define:nn { hyp / setup }
1393 {
1394     ,unicode .code:n = {}
1395     ,pdfencoding .code:n = {}
1396     ,pdfversion .code:n =
1397     {
1398         \msg_warning:nn { hyp }{ pdfversion-disabled }
1399     }
1400 }
1401 %

```

### 12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

`verbose` (*setup key*)

`debug` (*setup key*)

`draft` (*setup key*)

`final` (*setup key*)

```

1402 \keys_define:nn { hyp / setup }
1403 {
1404     ,verbose .choice:
1405     ,verbose / true .code:n = { \Hy@verbosetrue}

```

```

1406 ,verbose / false .code:n = { \Hy@verbosefalse}
1407 ,verbose .default:n = {true}
1408 ,debug .meta:n = {verbose=#1}
1409 ,debug .default:n = {true}
1410 }
1411 \keys_define:nn { hyp / setup }
1412 {
1413 ,draft .code:n =
1414 {
1415 \Hy@drafttrue
1416 \PassOptionsToPackage{draft}{bookmark}
1417 }
1418 ,final .code:n =
1419 {
1420 \Hy@finaltrue
1421 \PassOptionsToPackage{final}{bookmark}
1422 }
1423 }

extension (setup key)
hypertextnames (setup key) 1424 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1425 {
pageanchor (setup key) 1426 ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1427 ,extension .initial:n= pdf
linktocpage (setup key) 1428 ,hypertextnames .choice:
plainpages (setup key) 1429 ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1430 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1431 ,hypertextnames .default:n = {true}
1432 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1433 ,localanchorname .choice:
1434 ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1435 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1436 ,localanchorname .default:n = {true}
1437 ,naturalnames .choice:
1438 ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1439 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1440 ,naturalnames .default:n = {true}
1441 ,pageanchor .choice:
1442 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1443 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1444 ,pageanchor .default:n = {true}
1445 ,plainpages .choice:
1446 ,plainpages / true .code:n = { \Hy@plainpagestrue}
1447 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1448 ,plainpages .default:n = {true}
1449 }
1450
1451 \keys_define:nn { hyp / setup }
1452 {
1453 ,linktoc .choices:nn = { none, section, all, page }
1454 {
1455 \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1456 }
1457 ,linktoc / unknown .code:n =

```

```

1458     {
1459         \msg_warning:nxxxx { hyp } { unknown-choice }
1460         { linktoc }
1461         { none, section, all, page }
1462         { \exp_not:n {#1} }
1463     }
1464     ,linktocpage .choice:
1465     ,linktocpage / true .meta:n = {linktoc=page}
1466     ,linktocpage / false .meta:n = {linktoc=section}
1467     ,linktocpage .default:n = true
1468 }
1469

```

link (*setup key*) This booleans allow to disable the link types.

```

url (setup key) 1470 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1471 {
menu (setup key) 1472 \keys_define:nn { hyp / setup }
run (setup key) 1473 {
1474     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1475 }
1476 }

```

```

1477 \keys_define:nn { hyp / setup }
1478 {
1479     ,baseurl .code:n =
1480     {
1481         \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1482         \tl_if_empty:NTF \l__hyp_tmpa_tl
1483         {
1484             \pdfmanagement_remove:nn {Catalog} { URI }
1485         }
1486         {
1487             \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1488         }
1489     }
1490     %only false does something ...
1491     ,bookmarks .choice:
1492     ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1493     ,bookmarks / true .code:n = {}
1494     ,bookmarks .default:n = {true}
1495     ,bookmarksnumbered .choice:
1496     ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1497     ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1498     ,bookmarksnumbered .default:n = {true}
1499     ,bookmarksopen .choice:
1500     ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1501     ,bookmarksopen / true .code:n = { \Hy@bookmarksopentru }
1502     ,bookmarksopen .default:n = {true}
1503     ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1504     ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1505     ,pdfcenterwindow .choice:
1506     ,pdfcenterwindow / false .code:n =
1507     {
1508         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }

```

```

1509     }
1510 ,pdfcenterwindow / true .code:n =
1511     {
1512         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1513     }
1514 ,pdfcenterwindow / .code:n =
1515     {
1516         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1517     }
1518 ,pdfcenterwindow / unknown .code:n =
1519     {
1520         \msg_warning:nxxx { hyp } { no-bool }
1521         { pdfcenterwindow }
1522         { \exp_not:n {#1} }
1523     }
1524 ,pdfcenterwindow .default:n = true
1525 ,pdfdirection .choice:
1526 ,pdfdirection / L2R .code:n =
1527     {
1528         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1529     }
1530 ,pdfdirection / R2L .code:n =
1531     {
1532         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1533     }
1534 ,pdfdirection / .code:n =
1535     {
1536         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1537     }
1538 ,pdfdirection / unknown .code:n =
1539     {
1540         \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1541         { pdfdirection }
1542         { L2R , R2L }
1543         { \exp_not:n {#1} }
1544     }
1545 ,pdfdisplaydoctitle .choice:
1546 ,pdfdisplaydoctitle / false .code:n =
1547     {
1548         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1549     }
1550 ,pdfdisplaydoctitle / true .code:n =
1551     {
1552         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1553     }
1554 ,pdfdisplaydoctitle .default:n = true
1555 ,pdfduplex .choices:nn =
1556 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1557 {
1558     \pdf_version_compare:NnTF > {1.6}
1559     {
1560         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1561         { PrintDuplex } { /#1 }
1562     }

```

```

1563     {
1564         \msg_warning:nxxx
1565         {hyp}
1566         {ignore-deprecated-or-unknown-option-in-pdf-version}
1567         {pdfduplex}
1568         {\pdf_version:}
1569     }
1570 }%
1571 ,pdfduplex / .code:n =
1572 {
1573     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1574 }
1575 ,pdfduplex / unknown .code:n =
1576 {
1577     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1578     { pdfduplex }
1579     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1580     { \exp_not:n {#1} }
1581 }
1582 ,pdffitwindow .choice:
1583 ,pdffitwindow / false .code:n =
1584 {
1585     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1586 }
1587 ,pdffitwindow / true .code:n =
1588 {
1589     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1590 }
1591 ,pdffitwindow / .code:n =
1592 {
1593     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1594 }
1595 ,pdffitwindow .default:n = true
1596 ,pdffitwindow / unknown .code:n =
1597 {
1598     \msg_warning:nxxx { hyp } { no-bool }
1599     { pdffitwindow }
1600     { \exp_not:n {#1} }
1601 }
1602 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1603 ,pdflinkmargin .initial:n = {1pt}
1604 ,pdfmenubar .choice:
1605 ,pdfmenubar / true .code:n =
1606 {
1607     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1608 }
1609 ,pdfmenubar / false .code:n =
1610 {
1611     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1612     { HideMenubar } { true }
1613 }
1614 ,pdfmenubar / .code:n =
1615 {
1616     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }

```

```

1617     }
1618     ,pdfmenubar .default:n = true
1619     ,pdfmenubar / unknown .code:n =
1620     {
1621         \msg_warning:nxxx { hyp } { no-bool }
1622         { pdfmenubar }
1623         { \exp_not:n {#1} }
1624     }
1625     ,pdfnewwindow .choice:
1626     ,pdfnewwindow / true .code:n =
1627     {
1628         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1629         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1630     }
1631     ,pdfnewwindow / false .code:n =
1632     {
1633         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1634         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1635     }
1636     ,pdfnewwindow / .code:n =
1637     {
1638         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1639         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1640     }
1641     ,pdfnonfullscreenpagemode .choices:nn =
1642     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1643     {
1644         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1645         { NonFullScreenPageMode } {/#1}
1646     }
1647     ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1648     {
1649         \pdf_version_compare:NnTF < {1.6}
1650         {
1651             %message
1652         }
1653         {
1654             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1655             {NonFullScreenPageMode}{/UseAttachments}
1656         }
1657     }
1658     ,pdfnonfullscreenpagemode / .code:n =
1659     {
1660         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1661     }
1662     ,pdfnonfullscreenpagemode / unknown .code:n =
1663     {
1664         \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1665         { pdfnonfullscreenpagemode }
1666         { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1667         { \exp_not:n {#1} }
1668     }
1669     ,pdfnumcopies .code:n =
1670     {

```

```

1671     \pdf_version_compare:NnTF > {1.6}
1672     {
1673       \tl_if_empty:nTF {#1}
1674       {
1675         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1676       }
1677       {
1678         \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1679         {NumCopies}{#1}
1680       }
1681     }
1682     {
1683       \msg_warning:nxxx
1684       {hyp}
1685       {ignore-deprecated-or-unknown-option-in-pdf-version}
1686       {pdfnumcopies}
1687       {\pdf_version:}
1688     }
1689   }
1690 ,pdfpagelayout .choices:nn =
1691   { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1692   { \pdfmanagement_add:nxx {Catalog} { PageLayout }{ /#1 } }
1693 ,pdfpagelayout / .code:n =
1694   { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1695 ,pdfpagelayout / unknown .code:n =
1696   {
1697     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1698     { pdfpagelayout }
1699     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1700     { \exp_not:n {#1} }
1701   }
1702 ,pdfpagemode .choices:nn =
1703   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1704   { \pdfmanagement_add:nxx {Catalog} { PageMode }{ /#1 } }
1705 ,pdfpagemode / UseAttachments .code:n =
1706   {
1707     \pdf_version_compare:NnTF > {1.5}
1708     {
1709       \pdfmanagement_add:nxx {Catalog} { PageMode }{ /UseAttachments }
1710     }
1711     {
1712       \msg_warning:nxxx
1713       {hyp}
1714       {ignore-deprecated-or-unknown-value-in-pdf-version}
1715       {UseAttachments}
1716       {\pdf_version:}
1717     }
1718   }
1719 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1720 ,pdfpagemode / unknown .code:n =
1721   {
1722     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1723     { pdfpagemode }
1724     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }

```

```

1725         { \exp_not:n {#1} }
1726     }
1727 ,pdfpagescrop .code:n =
1728     {
1729     \tl_if_empty:nTF {#1} %or blank?
1730     {
1731     \pdfmanagement_remove:nn {Pages} { CropBox }
1732     }
1733     {
1734     \pdfmanagement_add:nxx {Pages} { CropBox } { [#1] }
1735     }
1736     }
1737 ,pdfpicktraybypdfsize .choice:
1738 ,pdfpicktraybypdfsize / true .code:n =
1739     {
1740     \pdf_version_compare:NnTF > {1.6}
1741     {
1742     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1743     { PickTrayByPDFSize } { true }
1744     }
1745     {
1746     \msg_warning:nxxx
1747     {hyp}
1748     {ignore-deprecated-or-unknown-option-in-pdf-version}
1749     {pdfpicktraybypdfsize}
1750     {\pdf_version:}
1751     }
1752     }
1753 ,pdfpicktraybypdfsize / false .code:n =
1754     {
1755     \pdf_version_compare:NnTF > {1.6}
1756     {
1757     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1758     { PickTrayByPDFSize } { false }
1759     }
1760     {
1761     \msg_warning:nxxx
1762     {hyp}
1763     {ignore-deprecated-or-unknown-option-in-pdf-version}
1764     {pdfpicktraybypdfsize}
1765     {\pdf_version:}
1766     }
1767     }
1768 ,pdfpicktraybypdfsize / .code:n =
1769     {
1770     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1771     }
1772 ,pdfpicktraybypdfsize / unknown .code:n =
1773     {
1774     \msg_warning:nxxx { hyp } { no-bool }
1775     { picktraybypdfsize }
1776     { \exp_not:n {#1} }
1777     }
1778 ,pdfprintarea .choices:nn =

```



```

1779 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1780 {
1781   \pdf_version_compare:NnTF < {2.0}
1782   {
1783     \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1784     { PrintArea } { /#1 }
1785   }
1786   {
1787     \msg_warning:nxxx
1788     {hyp}
1789     {ignore-deprecated-or-unknown-option-in-pdf-version}
1790     {pdfprintarea}
1791     {\pdf_version:}
1792   }
1793 }%
1794 ,pdfprintarea / .code:n =
1795 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1796 ,pdfprintarea / unknown .code:n =
1797 {
1798   \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1799   { pdfprintarea }
1800   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1801   { \exp_not:n {#1} }
1802 }
1803 ,pdfprintclip .choices:nn =
1804 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1805 {
1806   \pdf_version_compare:NnTF < {2.0}
1807   {
1808     \pdfmanagement_add:nxx {Catalog / ViewerPreferences }
1809     { PrintClip } { /#1 }
1810   }
1811   {
1812     \msg_warning:nxxx
1813     {hyp}
1814     {ignore-deprecated-or-unknown-option-in-pdf-version}
1815     {pdfprintclip}
1816     {\pdf_version:}
1817   }
1818 }%
1819 ,pdfprintclip / .code:n =
1820 {
1821   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1822 }
1823 ,pdfprintclip / unknown .code:n =
1824 {
1825   \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1826   { pdfprintclip }
1827   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1828   { \exp_not:n {#1} }
1829 }
1830 ,pdfprintpagerange .code:n =
1831 {
1832   \pdf_version_compare:NnTF > {1.6}

```

```

1833     {
1834         \tl_if_empty:nTF { #1}
1835         {
1836             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1837             { PrintPageRange }
1838         }
1839         {
1840             \pdfmanagement_add:nx {Catalog / ViewerPreferences }
1841             {PrintPageRange}{[#1]}
1842         }
1843     }
1844     {
1845         \msg_warning:nxxx
1846         {hyp}
1847         {ignore-deprecated-or-unknown-option-in-pdf-version}
1848         {pdfprintpagerange}
1849         {\pdf_version:}
1850     }
1851 }
1852 ,pdfprintscaling .choices:nn =
1853 { None, AppDefault }
1854 {
1855     \pdf_version_compare:NnTF > {1.5}
1856     {
1857         \pdfmanagement_add:nx {Catalog / ViewerPreferences }
1858         { PrintScaling } { /#1 }
1859     }
1860     {
1861         \msg_warning:nxxx
1862         {hyp}
1863         {ignore-deprecated-or-unknown-option-in-pdf-version}
1864         {pdfprintscaling}
1865         {\pdf_version:}
1866     }
1867 }%
1868 ,pdfprintscaling / .code:n =
1869 {
1870     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1871 }
1872 ,pdfprintscaling / unknown .code:n =
1873 {
1874     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1875     { pdfprintarea }
1876     { None, AppDefault }
1877     { \exp_not:n {#1} }
1878 }
1879 ,pdfremotestartview .code:n =
1880 {
1881     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1882     \exp_args:NNV
1883     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1884     {
1885         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1886     }

```

```

1887     {
1888         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1889         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1890     }
1891 }
1892 ,pdfremotestartview .initial:n = {Fit}
1893 % pdfstartpage is special as it shares code with pdfstartview
1894 ,pdfstartpage .code:n =
1895 {
1896     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1897     \bool_if:nTF
1898     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl }
1899     {
1900         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1901     }
1902     {
1903         \pdfmanagement_add:nxx {Catalog} { OpenAction }
1904         {
1905             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartpage_tl]
1906         }
1907     }
1908 }
1909 ,pdfstartpage .initial:n = 1
1910 ,pdfstartview .code:n =
1911 {
1912     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1913     \exp_args:NNV
1914     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_seq {1}
1915     {
1916         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1917     }
1918     {
1919         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1920         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1921     }
1922     \bool_if:nTF
1923     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl }
1924     {
1925         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1926     }
1927     {
1928         \pdfmanagement_add:nxx {Catalog} { OpenAction }
1929         {
1930             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartpage_tl]
1931         }
1932     }
1933 }
1934 ,pdfstartview .initial:n = Fit
1935 ,pdftoolbar .choice:
1936 ,pdftoolbar / true .code:n =
1937 {
1938     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1939 }
1940 ,pdftoolbar / false .code:n =

```

```

1941     {
1942     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1943     { HideToolbar } { true }
1944     }
1945 ,pdftoolbar / true .code:n =
1946     {
1947     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1948     }
1949 ,pdftoolbar .default:n = true
1950 ,pdftoolbar / unknown .code:n =
1951     {
1952     \msg_warning:nxxx { hyp } { no-bool }
1953     { pdftoolbar }
1954     { \exp_not:n {#1} }
1955     }
1956 % pdfview see below.
1957 ,pdfviewarea .choices:nn =
1958     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1959     {
1960     \pdf_version_compare:NnTF < {2.0}
1961     {
1962     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1963     { ViewArea } { /#1 }
1964     }
1965     {
1966     \msg_warning:nxxx
1967     {hyp}
1968     {ignore-deprecated-or-unknown-option-in-pdf-version}
1969     {pdfviewarea}
1970     {\pdf_version:}
1971     }
1972     }%
1973 ,pdfviewarea / .code:n =
1974     {
1975     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1976     }
1977 ,pdfviewarea / unknown .code:n =
1978     {
1979     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1980     { pdfviewarea }
1981     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1982     { \exp_not:n {#1} }
1983     }
1984 ,pdfviewclip .choices:nn =
1985     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1986     {
1987     \pdf_version_compare:NnTF < {2.0}
1988     {
1989     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1990     { ViewClip } { /#1 }
1991     }
1992     {
1993     \msg_warning:nxxx
1994     {hyp}

```

```

1995         {ignore-deprecated-or-unknown-option-in-pdf-version}
1996         {pdfviewclip}
1997         {\pdf_version:}
1998     }
1999 }%
2000 ,pdfviewclip / .code:n =
2001 {
2002     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2003 }
2004 ,pdfviewclip / unknown .code:n =
2005 {
2006     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
2007     { pdfviewclip }
2008     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2009     { \exp_not:n {#1} }
2010 }
2011 ,pdfwindowui .choice:
2012 ,pdfwindowui / true .code:n =
2013 {
2014     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2015 }
2016 ,pdfwindowui / false .code:n =
2017 {
2018     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2019     { HideWindowUI } { true }
2020 }
2021 ,pdfwindowui / .code:n =
2022 {
2023     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2024 }
2025 ,pdfwindowui / unknown .code:n =
2026 {
2027     \msg_warning:nnxx { hyp } { no-bool }
2028     { pdfwindowui }
2029     { \exp_not:n {#1} }
2030 }
2031 ,pdfwindowui .default:n = true
2032 }

```

`pdfview` (*setup key*) Destination keys. `pdfview` is a bit more complicated so extra.

```

2033 \keys_define:nn { hyp / setup }
2034 {
2035     ,pdfview .code:n =
2036     {
2037         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2038         \str_case_e:nmF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2039         {
2040             { xyz }
2041             {
2042                 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2043                 {
2044                     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2045                     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2046                     {

```

```

2047         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2048     }
2049     {
2050         \tl_set:Nx \l__hyp_dest_pdfview_tl
2051         {
2052             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2053         }
2054     }
2055 }
2056 {
2057     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2058 }
2059 }
2060 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2061 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2062 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2063 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2064 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2065 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2066 { fitr }
2067 {
2068     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2069     {
2070         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2071     }
2072     {
2073         %ensure 4 values ...
2074         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2075         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2076         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2077         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2078         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2079         {
2080             \fp_eval:n
2081             {
2082                 round
2083                 (
2084                     abs
2085                     (
2086                         \seq_item:Nn\l__hyp_tmpa_seq{4}
2087                         -
2088                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2089                     ),
2090                     3
2091                 )
2092             }bp
2093         }{ }
2094         \box_set_dp:Nn \l__hyp_dest_box
2095         {
2096             \fp_eval:n
2097             {
2098                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2099             }bp
2100         }

```

```

2101             \box_set_ht:Nn \l__hyp_dest_box
2102             {
2103                 \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2104             }
2105         }
2106     }
2107 }
2108 {
2109     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2110     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2111 }
2112 }
2113 ,pdfview .initial:n = {xyz}
2114 }

```

## 12.3 “MetaData keys”

The following keys are relevant for the metadata: the info dictionary and the xmp-metadata.

`pdflang` (*setup key*) `pdflang` should be deprecated.

```

2115 \keys_define:nn { hyp / setup }
2116 {
2117     ,pdflang .code:n =
2118     {
2119         \tl_if_empty:nTF { #1 }
2120         {
2121             \pdfmanagement_remove:nn {Catalog} { Lang }
2122         }
2123         {
2124             \pdfmanagement_add:nx {Catalog} { Lang } { (#1) }
2125         }
2126         \__hyp_store_metadata:nn {pdflang}{#1}
2127     }
2128 }

```

### 12.3.1 “info keys”

`pdfauthor` (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use them. Creator and Producer can't be removed with the pdfmanagement, but we allow `pdfcreator` (*setup key*) to set an empty value.

```

pdfsubject (setup key) 2129 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfproducer (setup key) 2130 {
pdfkeywords (setup key) 2131     \keys_define:nn { hyp / setup }
2132     {
2133         pdf#1 .code:n =
2134         {
2135             \tl_if_blank:nTF {##1}
2136             {
2137                 \str_case:nnF { #1 }
2138                 {
2139                     {creator}
2140                 }

```

```

2141         \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2142         \pdfmanagement_add:nnx {Info}{Creator}{()}
2143     }
2144     {producer}
2145     {
2146         \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2147         \pdfmanagement_add:nnx {Info}{Producer}{()}
2148     }
2149 }
2150 {
2151     \pdfmanagement_remove:nn {Info}{#2}
2152 }
2153 }
2154 {
2155     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2156     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2157     {
2158         \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2159     }
2160 }
2161 \__hyp_store_metadata:nn {pdf#1}{##1}
2162 }
2163 }
2164 \keys_define:nn { hyp / info }
2165 {
2166     #2 .code:n =
2167     {
2168         \tl_if_blank:nTF {##1}
2169         {
2170             \pdfmanagement_remove:nn {Info}{#2}
2171         }
2172         {
2173             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2174             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2175             {
2176                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2177             }
2178         }
2179         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2180     }
2181     ,unknown .code:n =
2182     {
2183         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2184         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2185         {
2186             \exp_args:Nno
2187             \pdfmanagement_add:nnx {Info}
2188             { \l_keys_key_str } {\l__hyp_tmpa_str}
2189         }
2190     }
2191 }
2192 }
2193 \__hyp_setup_info_key:nn {author} {Author}
2194 \__hyp_setup_info_key:nn {title} {Title}

```



```

2195 \_hyp_setup_info_key:nn {producer} {Producer}
2196 \_hyp_setup_info_key:nn {creator} {Creator}
2197 % ignored key: addtopdfcreator
2198 \_hyp_setup_info_key:nn {subject} {Subject}
2199 \_hyp_setup_info_key:nn {keywords} {Keywords}

```

pdfcreationdate (setup key) These keys are not really needed. We store them too in the container. CreationDate and pdfmoddate (setup key) ModDate should not use the hex encoding.

```

2200 \cs_new_protected:Npn \_hyp_setup_info_date_key:nn #1 #2
2201 {
2202   \keys_define:nn { hyp / setup }
2203   {
2204     pdf#1 .code:n =
2205     {
2206       \tl_if_blank:nTF {##1}
2207       {
2208         \pdfmanagement_remove:nn {Info}{#2}
2209       }
2210       {
2211         \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2212       }
2213       \_hyp_store_metadata:nn {pdf#1}{##1}
2214     }
2215   }
2216   \keys_define:nn { hyp / info }
2217   {
2218     #2 .code:n =
2219     {
2220       \tl_if_blank:nTF {##1}
2221       {
2222         \pdfmanagement_remove:nn {Info}{#2}
2223       }
2224       {
2225         \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2226       }
2227       \exp_args:Nx \_hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2228     }
2229   }
2230 }
2231
2232 \_hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2233 \_hyp_setup_info_date_key:nn {moddate} {ModDate}

```

pdftrapped (setup key) Trapped is a bit curious, it has an value unknown, and one can't suppress it ...

```

2234 \keys_define:nn { hyp / setup }
2235 {
2236   ,pdftrapped .code:n =
2237   {
2238     \exp_args:Nne
2239     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2240   }
2241   ,pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2242   {
2243     \pdfmanagement_add:nnx {Info}{Trapped}

```

```

2244         {/
2245         \str_uppercase:f { \str_head:n { #1 } }
2246         \str_lowercase:f { \str_tail:n { #1 } }
2247         }
2248     \__hyp_store_metadata:nx {pdftrapped}
2249     {
2250         \str_uppercase:f { \str_head:n { #1 } }
2251         \str_lowercase:f { \str_tail:n { #1 } }
2252     }
2253 }
2254 ,_pdftrapped / unknown .code:n =
2255 {
2256     \msg_warning:nxxxx { hyp } { unknown-choice }
2257     { pdftrapped }
2258     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2259     { \exp_not:n {#1} }
2260 }
2261 }

```

`pdfinfo (setup key)` `pdfinfo` allows to set the info keys with keyval ...

```

2262 \keys_define:nn { hyp / setup }
2263 {
2264     pdfinfo .code:n =
2265     {
2266         \keys_set:nn { hyp / info } { #1 }
2267     }
2268 }

```

Now we set some default values

```

2269 \keys_set:nn { hyp / setup} {pdfcreator = LaTeX-with-hyperref}
2270 \keys_set:nn { hyp / setup} {pdfauthor = }
2271 \keys_set:nn { hyp / setup} {pdftitle = }
2272 \keys_set:nn { hyp / setup} {pdfsubject = }

```

## 12.4 hyperxmp keys

`hyperxmp` defines lots of keys for `\hypersetup`. They now longer work with this driver. So we provide them, but they are only stored as metadata:

```

2273 \clist_map_inline:nn
2274 {
2275     ,pdfcopyright
2276     ,pdftype
2277     ,pdflicenseurl
2278     ,pdfauthortitle
2279     ,pdfcaptionwriter
2280     ,pdfmetalang
2281     ,pdfapart
2282     ,pdfaconformance
2283     ,pdfuapart
2284     ,pdfxstandard
2285     ,pdfsource
2286     ,pdfdocumentid
2287     ,pdfinstanceid

```

```

2288 ,pdfversionid
2289 ,pdfrendition
2290 ,pdfpublication
2291 ,pdfpubtype
2292 ,pdfbytes
2293 ,pdfnumpages
2294 ,pdfissn
2295 ,pdfeissn
2296 ,pdfisbn
2297 ,pdfbookedition
2298 ,pdfpublisher
2299 ,pdfvolumenum
2300 ,pdfissuenum
2301 ,pdfpagerange
2302 ,pdfdoi
2303 ,pdfurl
2304 ,pdfidentifier
2305 ,pdfsubtitle
2306 ,pdfpubstatus
2307 ,pdfcontactaddress
2308 ,pdfcontactcity
2309 ,pdfcontactregion
2310 ,pdfcontactpostcode
2311 ,pdfcontactcountry
2312 ,pdfcontactphone
2313 ,pdfcontactemail
2314 ,pdfcontacturl
2315 ,pdfdate
2316 }
2317 {
2318   \keys_define:nn { hyp / setup }
2319   {
2320     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2321   }
2322 }
2323
2324 \str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
2325 {
2326   \exp_args:Nnx
2327   \keys_set:nn {hyp / setup}
2328   {
2329     ,pdfapart      = \pdfmeta_standard_item:n {level}
2330     ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
2331   }
2332 }

```

## 12.5 Transitions

pdfpageduration sets the duration a page is shown in full screen mode.

```

2333 \keys_define:nn { hyp / setup }
2334 {
2335   pdfpageduration .code:n =
2336   {
2337     \tl_if_blank:nTF { #1 }

```

```

2338     {
2339     \pdfmanagement_remove:nn {Page}{Dur}
2340     }
2341     {
2342     \pdfmanagement_add:nnn {Page}{Dur}{#1}
2343     }
2344   }
2345 }

```

Transition settings are used by (some) pdf viewers when presenting a pdf in full screen mode. They are added to the page settings and describe the transition from the previous page to current page. Transition setting can be set in the preamble for all pages or in the document for the current and the following pages. Due to the asynchronous page breaking one has to be careful to set it on the right page, e.g. only after a `\newpage`. The generic driver uses a different syntax than the other `hyperref` drivers: various transition options can be set by a keyval syntax in the value of `pdfpagetransition`. A typical setting looks e.g. like this

```

\hypersetup{pdfpagetransition={style=Fly,duration=2,direction=90,opaque=false}}

```

style	one of Split, Blinds, Box, Dissolve, Glitter, R, Fly, I, Uncover, Fade
duration	a number, describes the duration of the transition
direction	H (horizontal, only Split, B), V (vertical, only Split, B), 0 (left to right, only Wipe), 90 (bottom to top, only Wipe), 180 (right to left, only Wipe), 270 (top to bottom, only Wipe), 315 (top left to bottom, only Wipe), None (only Fly)
motion	one of I, O, only relevant for Fly and Fly
scale	a number, only relevant for Fly
opaque	true or false, only relevant for Fly

The keys allowed in the argument of `pdfpagetransition` are

```

2346 \keys_define:nn { hyp / setup }
2347 {
2348   pdfpagetransition .code:n =
2349   {
2350     \tl_if_blank:nTF {#1}
2351     {
2352       \pdfmanagement_remove:nn {Page}{Trans}
2353     }
2354     {
2355       \group_begin:
2356       \keys_set:nn { hyp / trans }{style=R,#1}
2357       \pdf_object_unnamed_write:nx { dict }
2358       {
2359         \pdfdict_use:n {l__hyp_page/Trans}
2360       }
2361       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2362       \group_end:

```

```

2363     }
2364   }
2365 }
2366 \keys_define:nn { hyp / trans }
2367 {
2368   ,style .choices:nn =
2369     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2370     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2371   ,style / unknown .code:n =
2372     {
2373       \msg_warning:nxxx { hyp } { unknown-choice }
2374       { trans / style }
2375       { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2376       { \exp_not:n {#1} }
2377     }
2378   ,duration .code:n =
2379     {
2380       \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2381     }
2382   ,direction .choices:nn =
2383     {H,V}
2384     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2385   ,direction .choices:nn =
2386     {0,90,180,270,315}
2387     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2388   ,direction / None .code:n =
2389     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2390   ,direction / unknown .code:n =
2391     {
2392       \msg_warning:nxxx { hyp } { unknown-choice }
2393       { trans / direction }
2394       {
2395         H~(horizontal,~only~Split,~Blinds),
2396         V~(vertical,~only~Split,~Blinds),
2397         0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2398         90~(bottom~to~top,~only~Wipe),
2399         180~(right~to~left,~only~Wipe),
2400         270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2401         315~(top~left~to~bottom,~only~Glitter),
2402         None~(only~Fly)
2403       }
2404       { \exp_not:n {#1} }
2405     }
2406   ,motion .choices:nn =
2407     {I,0}
2408     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2409   ,motion / unknown .code:n =
2410     {
2411       \msg_warning:nxxx { hyp } { unknown-choice }
2412       { trans / motion }
2413       { I~(inwards) , 0~(outwards) }
2414       { \exp_not:n {#1} }
2415     }
2416   ,scale .code:n =

```

```

2417     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2418     ,opaque .choices:nn = {true,false}
2419     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2420     ,opaque / unknown .code:n =
2421     {
2422         \msg_warning:nnxxx { hyp } { unknown-choice }
2423         { trans / B }
2424         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2425         { \exp_not:n {#1} }
2426     }
2427     % try to set unknown keys as style
2428     ,unknown .code:n =
2429     {
2430         % warning ...
2431         \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2432     }
2433 }

```

Finally we process the package option list, to get most keys working

```

2434 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2435 \NewDocumentCommand \MakeFieldObject { m m }
2436 {
2437     \pdfxform_new:nnn { #2 }{ } { #1 }
2438 }
2439
2440
2441 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2442 \prop_new:N \g__hyp_AcroForm_Fields_prop
2443
2444 \let\HyField@afields\ltx@empty
2445 \let\HyField@cofields\ltx@empty
2446 %% UF test for old pdftex removed
2447 %\let\HyField@AuxAddToFields\ltx@gobble
2448 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2449 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2450
2451 % the value doesn't matter, but with a prop we avoid duplicates and it is
2452 % clearly faster than removing them from a sequence
2453 \def\HyField@AuxAddToFields#1
2454 {
2455     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2456 }%
2457
2458 %fields with empty key get a value too -- lets hope that
2459 %this give the expected behaviour
2460 \def\HyField@AuxAddToCoFields #1 #2
2461 {
2462     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2463 }
2464
2465 \Hy@AtBeginDocument
2466 {
2467     \if@filesw

```

```

2468     \immediate\write\@mainaux{%
2469       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2470     }%
2471     \immediate\write\@mainaux{%
2472       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2473     }%
2474     \fi
2475     \let\HyField@AfterAuxOpen\@firstofone
2476   }%
2477
2478 \def\HyField@AddToFields
2479 {
2480   \exp_args:Nx\HyField__hypAddToFields
2481   {
2482     \pdfannot_box_ref_last:
2483   }
2484   \ifx\Fld@calculate@code\ltx@empty
2485   \else
2486     \begingroup
2487     \Hy@safe@activestrue
2488     \edef\Hy@temp{%
2489       \endgroup
2490       \if@filesw
2491         \write\@mainaux
2492         {
2493           \string\HyField@AuxAddToCoFields
2494           {
2495             \Fld@calculate@sortkey
2496           }
2497           {
2498             \pdfannot_box_ref_last:
2499           }
2500         }
2501       \fi
2502     }%
2503     \Hy@temp
2504   \fi
2505 }%
2506
2507 \def\HyField__hypAddToFields#1{
2508   \HyField@AfterAuxOpen{%
2509     \if@filesw
2510       \write\@mainaux{%
2511         \string\HyField@AuxAddToFields{#1}%
2512       }%
2513     \fi
2514   }%
2515 }%
2516
2517 \ExplSyntaxOff
2518 \ExplSyntaxOn
2519
2520 \def\@Form[#1]
2521 {

```

```

2522 \kvsetkeys{Form}{#1}
2523 \pdf@ifdraftmode{}
2524 {
2525   \Hy@FormObjects
2526   \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2527   {
2528     \pdfmanagement_add:nx { Catalog / AcroForm } { Fields }{##1}
2529     %\pdfmanagement_show:n { Catalog / AcroForm }
2530   }
2531   \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2532   {
2533     \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2534     {
2535       \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2536     }
2537     \seq_sort:Nn \l__hyp_tmpa_seq
2538     {
2539       \str_compare:nNnTF {##1} > {##2}
2540       { \sort_return_swapped: }
2541       { \sort_return_same: }
2542     }
2543     \seq_map_inline:Nn \l__hyp_tmpa_seq
2544     {
2545       \pdfmanagement_add:nx { Catalog / AcroForm }
2546       { CO }
2547       {
2548         \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2549       }
2550     }
2551   }
2552   \pdfmanagement_add:nx {Catalog / AcroForm/DR/Font }
2553   {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2554   \pdfmanagement_add:nx {Catalog / AcroForm/DR/Font }
2555   {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2556   \pdfmanagement_add:nx {Catalog /AcroForm}
2557   {DA}{(/Helv~10~Tf~0~g)}
2558   \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2559   {
2560     \legacy_if:nT { HyField@NeedAppearances }
2561     {
2562       \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2563     }
2564   }
2565   {
2566     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2567   }
2568 }
2569 }
2570 \ExplSyntaxOff
2571 \let\@endForm\ltx@empty
2572 \let\HyAnn@AbsPageLabel\ltx@empty
2573 \let\Fld@pageobjref\ltx@empty
2574
2575 \ExplSyntaxOn

```



```

2576 \newcount\HyAnn@Count
2577 \HyAnn@Count=\ltx@zero
2578 \def\HyAnn@AbsPageLabel
2579 {
2580   \global\advance\HyAnn@Count by\ltx@one
2581   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2582   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2583   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2584   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2585   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2586 }%
2587 \def\Fld@pageobjref
2588 {
2589   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2590   {
2591     /P~\pdf_pageobject_ref:n
2592     {
2593       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2594     }
2595   }
2596 }
2597 \ExplSyntaxOff
2598 \ExplSyntaxOn
2599 %% check if the attr should be set through
2600 %% hooks.
2601 %% check if options are missing.
2602 \def\@TextField[#1]#2{% parameters, label
2603   \def\Fld@name{#2}%
2604   \let\Fld@default\ltx@empty
2605   \let\Fld@value\@empty
2606   \def\Fld@width{\DefaultWidthofText}%
2607   \def\Fld@height{%
2608     \ifFld@multiline
2609       \DefaultHeightofTextMultiline
2610     \else
2611       \DefaultHeightofText
2612     \fi
2613   }%
2614   \begingroup
2615     \expandafter\HyField@SetKeys\expandafter{%
2616       \DefaultOptionsofText,#1%
2617     }%
2618     \PDFForm@Name
2619     \HyField@FlagsText
2620     \ifFld@hidden\def\Fld@width{1sp}\fi
2621     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2622     \LayoutTextField{#2}{%
2623       \leavevmode
2624       \HyAnn@AbsPageLabel
2625       \Hy@escapeform\PDFForm@Text
2626       \pdfannot_box:nnnn
2627         {\Fld@width}
2628         {\Fld@height}
2629         {Opt} %is this correct?

```

```

2630         {\PDFForm@Text}
2631         \MakeTextField{\Fld@width}{\Fld@height}
2632         \HyField@AddToFields
2633     }%
2634 \endgroup
2635 }
2636 \providecommand\@curropt{}
2637 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2638     \def\Fld@name{#2}
2639     \let\Fld@default\relax
2640     \let\Fld@value\relax
2641     \def\Fld@width{\DefaultWidthofChoiceMenu}
2642     \def\Fld@height{\DefaultHeightofChoiceMenu}
2643     \begingroup
2644         \Fld@menulength=0 %
2645         \@tempdima\z@
2646         \clist_map_variable:nNn { #3 } \@curropt
2647         %\@for\@curropt:=#3\do
2648         {%
2649             \expandafter\Fld@checkequals\@curropt==\%
2650             \Hy@StepCount\Fld@menulength
2651             \settowidth{\@tempdimb}{\@currDisplay}%
2652             \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2653         }%
2654         \advance\@tempdima by~15\p@
2655         \begingroup
2656             \HyField@SetKeys{#1}
2657         \edef\x{\endgroup
2658             \noexpand\expandafter
2659             \noexpand\HyField@SetKeys
2660             \noexpand\expandafter{
2661                 \expandafter\noexpand\csname DefaultOptionsof%
2662                 \ifFld@radio
2663                     Radio%
2664                 \else
2665                     \ifFld@combo
2666                         \ifFld@popdown
2667                             PopdownBox%
2668                         \else
2669                             ComboBox%
2670                         \fi
2671                     \else
2672                         ListBox%
2673                     \fi
2674                 \fi
2675                 \endcsname
2676             }%
2677         }\x
2678         \HyField@SetKeys{#1}%
2679         \PDFForm@Name
2680         \ifFld@hidden\def\Fld@width{1sp}\fi
2681         \ifx\Fld@value\relax
2682             \let\Fld@value\Fld@default
2683         \fi

```

```

2684 \LayoutChoiceField{#2}{%
2685 \ifFld@radio
2686 \HyField@FlagsRadioButton
2687 \__hypRadio{#3}%
2688 \else
2689 \beginngroup
2690 \HyField@FlagsChoice
2691 \ifdim\Fld@width<\@tempdima
2692 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2693 \edef\Fld@width{\the\@tempdima}%
2694 \fi
2695 \ifFld@combo
2696 \else
2697 \@tempdima=\the\Fld@menulength\Fld@charsize
2698 \advance\@tempdima by-\Fld@borderwidth bp %
2699 \advance\@tempdima by-\Fld@borderwidth bp %
2700 \edef\Fld@height{\the\@tempdima}%
2701 \fi
2702 \__hypListBox{#3}%
2703 \endngroup
2704 \fi
2705 }%
2706 \endgroup
2707 }
2708
2709 \def\__hypRadio#1{%
2710 \Fld@listcount=0-%
2711 %\show\Fld@default
2712 \EdefEscapeName\Fld@default{\Fld@default}%
2713 \clist_map_variable:nNn { #1 } \@curropt
2714 %\@for\@curropt:=#1\do
2715 {%
2716 \expandafter\Fld@checkequals\@curropt==\%
2717 \EdefEscapeName\@currValue{\@currValue}%
2718 \Hy@StepCount\Fld@listcount
2719 \@currDisplay\space
2720 \leavevmode
2721 \HyAnn@AbsPageLabel
2722 \Hy@escapeform\PDFForm@Radio
2723 \pdfxform_if_exist:nF { __hyp_xform_Ding }
2724 {
2725 \pdfxform_new:nnn { __hyp_xform_Ding } {}
2726 {
2727 \group_begin:
2728 \fontfamily{pzd}
2729 \fontencoding{U}
2730 \fontseries{m}
2731 \fontshape{n}
2732 \selectfont
2733 \char123
2734 \group_end:
2735 }
2736 }
2737 \pdfannot_box:nnnx

```

```

2738     {\Fld@width}
2739     {\Fld@height}
2740     {Opt} %is this correct?
2741     {
2742         \PDFForm@Radio
2743         /AP
2744         <<
2745         /N
2746         <<
2747         /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2748         %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2749         >>
2750         >>
2751     }
2752     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2753     \int_compare:nNnT { \Fld@listcount} = { 1 }
2754     { \HyField@AddToFields }
2755     \c_space_tl % deliberate space between radio buttons
2756     % to do: --> should be configurable
2757 }%
2758 }
2759
2760 \newcount\Fld@listcount
2761 \def\__hypListBox#1
2762 {
2763     \HyField@PDFChoices{#1}
2764     \mode_leave_vertical:
2765     \HyAnn@AbsPageLabel
2766     \Hy@escapeform\PDFForm@List
2767     \pdf_link_user:nnn
2768     {widget} %perhaps we need more types??
2769     {\PDFForm@List}
2770     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2771     \HyField@AddToFields
2772 }
2773
2774
2775 \def\@PushButton[#1]#2{% parameters, label
2776     \def\Fld@name{#2}%
2777     \group_begin:
2778     \exp_args:No\HyField@SetKeys
2779     {
2780         \DefaultOptionsofPushButton,#1
2781     }
2782     \PDFForm@Name
2783     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2784     {
2785         \HyField@FlagsPushButton
2786         \legacy_if:nT {Fld@hidden}
2787         {
2788             \def\Fld@width{1sp}
2789         }
2790         \LayoutPushButtonField
2791         {

```

```

2792         \mode_leave_vertical:
2793         \HyAnn@AbsPageLabel
2794         \Hy@escapeform\PDFForm@Push
2795         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2796         \pdfannot_box:nnnn
2797         { \box_wd:N\l_tmpa_box}
2798         { \box_ht:N\l_tmpa_box}
2799         { \box_dp:N\l_tmpa_box} %is this correct?
2800         { \PDFForm@Push}
2801         { \box_use:N\l_tmpa_box}
2802         \HyField@AddToFields
2803     }
2804 }
2805 {
2806     \msg_error:nn { hyp }{ pdfa-no-push-button }
2807     \LayoutPushButtonField
2808     {
2809         \mode_leave_vertical:
2810         \MakeButtonField{#2}
2811     }
2812 }
2813 \group_end:
2814 }
2815
2816 \def\@Submit[#1]#2
2817 {
2818     \def\Fld@width {\DefaultWidthofSubmit}
2819     \def\Fld@height{\DefaultHeightofSubmit}
2820     \group_begin:
2821     \exp_args:No\HyField@SetKeys
2822     {
2823         \DefaultOptionsofSubmit,#1
2824     }
2825     \HyField@FlagsPushButton
2826     \HyField@FlagsSubmit
2827     \legacy_if:nT { Fld@hidden }
2828     {
2829         \def\Fld@width{1sp}
2830     }
2831     \mode_leave_vertical:
2832     \HyAnn@AbsPageLabel
2833     \Hy@escapeform\PDFForm@Submit
2834     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2835     \pdfxform_if_exist:nF
2836     { __hyp_xform_Submit }
2837     {
2838         \pdfxform_new:nnn { __hyp_xform_Submit }{-}
2839         {
2840             \fbox{\color_select:n{yellow}\textsf{Submit}}
2841         }
2842         \pdfxform_new:nnn { __hyp_xform_SubmitP }{-}
2843         {
2844             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2845         }

```

```

2846 }
2847 \pdfannot_box:nnnn
2848   {\box_wd:N\l_tmpa_box}
2849   {\box_ht:N\l_tmpa_box}
2850   {\box_dp:N\l_tmpa_box} %is this correct?
2851   {
2852     \PDFForm@Submit
2853     /AP<<
2854       /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2855       /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2856     >>
2857   }
2858   \HyField@AddToFields
2859   \box_use:N\l_tmpa_box
2860
2861 \group_end:
2862 }
2863
2864 \def\@Reset[#1]#2
2865 {
2866   \def\Fld@width {\DefaultWidthofReset}
2867   \def\Fld@height{\DefaultHeightofReset}
2868   \group_begin:
2869     \exp_args:No\HyField@SetKeys
2870     {
2871       \DefaultOptionsofReset,#1
2872     }
2873     \mode_leave_vertical:
2874     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2875     {
2876       \HyField@FlagsPushButton
2877       \legacy_if:nT { Fld@hidden }
2878         { \def\Fld@width{1sp} }
2879       \HyAnn@AbsPageLabel
2880       \Hy@escapeform\PDFForm@Reset
2881       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2882       \pdfannot_box:nnnn
2883         {\box_wd:N\l_tmpa_box}
2884         {\box_ht:N\l_tmpa_box}
2885         {\box_dp:N\l_tmpa_box} %is this correct?
2886         { \PDFForm@Reset }
2887       \HyField@AddToFields
2888       \box_use:N \l_tmpa_box
2889     }
2890     {
2891       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2892       \MakeButtonField{#2}
2893     }
2894   \group_end:
2895 }
2896
2897 \def\@CheckBox[#1]#2
2898   {% parameters, label
2899   \def\Fld@name{#2}

```

```

2900 \def\Fld@default{0}
2901 \group_begin:
2902   \def\Fld@width {\DefaultWidthofCheckBox}
2903   \def\Fld@height{\DefaultHeightofCheckBox}
2904   \exp_args:No\HyField@SetKeys
2905     {
2906       \DefaultOptionsofCheckBox,#1
2907     }
2908   \PDFForm@Name
2909   \HyField@FlagsCheckBox
2910   \legacy_if:nT { Fld@hidden }
2911     {
2912       \def\Fld@width{1sp}
2913     }
2914   \LayoutCheckField{#2}
2915     {
2916       \mode_leave_vertical:
2917       \HyAnn@AbsPageLabel
2918       \Hy@escapeform\PDFForm@Check
2919       \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2920         {
2921           \pdfxform_new:nnn
2922             {__hyp_xform_CheckMarkYes}{-}
2923             {
2924               \group_begin:
2925               \fontfamily{pzd}
2926               \fontencoding{U}
2927               \fontseries{m}
2928               \fontshape{n}
2929               \selectfont
2930               \char51
2931               \group_end:
2932             }
2933           \pdfxform_new:nnn
2934             {__hyp_xform_CheckMarkOff}{-}
2935             {
2936               \group_begin:
2937               \fontfamily{pzd}
2938               \fontencoding{U}
2939               \fontseries{m}
2940               \fontshape{n}
2941               \selectfont
2942               \phantom{\char51} %perhaps xetex needs some small glyph ..
2943               \group_end:
2944             }
2945           }
2946       \pdfannot_box:nnnn
2947         {\Fld@width}
2948         {\Fld@height}
2949         {Opt} %is this correct?
2950         {\PDFForm@Check}
2951       \HyField@AddToFields %check if this works with xelatex ...
2952     }
2953 \group_end:

```

```

2954 }
2955 \ExplSyntaxOff
2956
2957 %hm. Should a luatex driver use type1 fonts in fields????
2958 \ExplSyntaxOn
2959 \def\Hy@FormObjects
2960 {
2961   \pdf_object_new:nn {__hyp/Encoding/pdfdoc } { dict }
2962   \pdf_object_new:nn {__hyp/Font/ZaDb } { dict }
2963   \pdf_object_new:nn {__hyp/Font/Helv } { dict }
2964   \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
2965   {
2966     /Type/Encoding
2967     /Differences[
2968       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2969       /ring/tilde
2970       \c_space_tl
2971       39/quotesingle
2972       \c_space_tl
2973       96/grave %
2974       \iow_newline:
2975       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2976       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2977       /quotedblbase/quotedblleft/quotedblright/quoteleft
2978       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2979       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2980       \iow_newline:
2981       164/currency
2982       \c_space_tl
2983       166/brokenbar
2984       \c_space_tl
2985       168/dieresis/copyright/ordfeminine
2986       \c_space_tl
2987       172/logicalnot/.notdef/registered/macron/degree/plusminus
2988       /twosuperior/threesuperior/acute/mu
2989       \c_space_tl
2990       183/periodcentered/cedilla/onesuperior/ordmasculine
2991       \c_space_tl
2992       188/onequarter/onehalf/threequarters
2993       \iow_newline:
2994       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2995       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2996       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2997       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2998       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2999       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3000       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3001       /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
3002       /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3003       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3004     ]
3005   }
3006   \pdf_object_write:nn {__hyp/Font/ZaDb }
3007   {

```



```

3008         /Type/Font
3009         /Subtype/Type1
3010         /Name/ZaDb
3011         /BaseFont/ZapfDingbats
3012     }
3013 \pdf_object_write:nx {__hyp/Font/Helv }
3014 {
3015     /Type/Font
3016     /Subtype/Type1
3017     /Name/Helv
3018     /BaseFont/Helvetica
3019     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3020 }
3021 \global\let\Hy@FormObjects\relax
3022 }
3023 \ExplSyntaxOff
3024 \providecommand*{\Fld@pageobjref}{}
3025 \ifcsname pdf@escapestring\endcsname
3026 \def\Hy@escapeform#1{%
3027     \ifHy@pdfescapeform
3028     \let\Hy@escapestring\pdfescapestring
3029     \else
3030     \let\Hy@escapestring\@firstofone
3031     \fi
3032 }%
3033 \Hy@escapeform{}%
3034 \else
3035 \let\Hy@escapestring\@firstofone
3036 \def\Hy@escapeform#1{%
3037     \ifHy@pdfescapeform
3038     \def\Hy@escapestring##1{%
3039         \noexpand\Hy@escapestring{\noexpand##1}%
3040     }%
3041     \edef\Hy@temp{#1}%
3042     \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
3043     \def\Hy@escapestring##1{%
3044         \@ifundefined{Hy@esc@\string##1}{%
3045             ##1%
3046             \ThisShouldNotHappen
3047         }{%
3048             \csname Hy@esc@\string##1\endcsname
3049         }%
3050     }%
3051     \else
3052     \let\Hy@escapestring\@firstofone
3053     \fi
3054 }%
3055 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3056     \ifx\#3\%
3057     \else
3058     \expandafter
3059     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3060     \ltx@ReturnAfterFi{%
3061     \Hy__hypescapeform#3\@nil

```

```

3062     }%
3063     \fi
3064 }%
3065 \fi
3066 \def\PDFForm@Name{%
3067   \PDFForm__hypName\Fld@name
3068   \ifx\Fld@altname\relax
3069   \else
3070     \PDFForm__hypName\Fld@altname
3071   \fi
3072   \ifx\Fld@mappingname\relax
3073   \else
3074     \PDFForm__hypName\Fld@mappingname
3075   \fi
3076 }
3077 \def\PDFForm__hypName#1{%
3078   \begingroup
3079     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3080     \ifHy@unicode
3081       \Hy@unicodedefalse
3082     \fi
3083     \fi
3084     \pdfstringdef\Hy@gtemp#1%
3085   \endgroup
3086   \let#1\Hy@gtemp
3087 }
3088 \def\Fld@X@additionalactions{%
3089   \ifx\Fld@keystroke@code\@empty
3090   \else
3091     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3092   \fi
3093   \ifx\Fld@format@code\@empty
3094   \else
3095     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3096   \fi
3097   \ifx\Fld@validate@code\@empty
3098   \else
3099     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3100   \fi
3101   \ifx\Fld@calculate@code\@empty
3102   \else
3103     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3104   \fi
3105   \ifx\Fld@onfocus@code\@empty
3106   \else
3107     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3108   \fi
3109   \ifx\Fld@onblur@code\@empty
3110   \else
3111     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3112   \fi
3113   \ifx\Fld@onmousedown@code\@empty
3114   \else
3115     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%

```

```

3116 \fi
3117 \ifx\Fld@onmouseup@code\@empty
3118 \else
3119   /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3120 \fi
3121 \ifx\Fld@onenter@code\@empty
3122 \else
3123   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3124 \fi
3125 \ifx\Fld@onexit@code\@empty
3126 \else
3127   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3128 \fi
3129 }
3130 \ExplSyntaxOn
3131 \def\Fld@additionalactions
3132 {%
3133   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{-}
3134   {
3135     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3136     {/AA<<\Fld@X@additionalactions>>}
3137   }
3138 }
3139 \ExplSyntaxOff
3140 \def\Fld@annotnames{%
3141   /T(\Fld@name)%
3142   \ifx\Fld@altname\relax
3143   \else
3144     /TU(\Fld@altname)%
3145   \fi
3146   \ifx\Fld@mappingname\relax
3147   \else
3148     /TM(\Fld@mappingname)%
3149   \fi
3150 }
3151 \ExplSyntaxOn
3152 \def\PDFForm@Check
3153 {
3154   /Subtype/Widget
3155   ~\Fld@annotflags
3156   ~\Fld@pageobjref
3157   ~\Fld@annotnames
3158   /FT/Btn
3159   \Fld@flags
3160   /Q~\Fld@align
3161   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3162   /AP
3163   <<
3164     /N
3165     <<
3166       /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3167       /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3168     >>
3169   >>

```

```

3170 /MK<<
3171   \int_compare:nNnF {\Fld@rotation}={0}
3172   {
3173     /R~\Fld@rotation
3174   }
3175   \tl_if_empty:NF\Fld@bordercolor
3176   {
3177     /BC[\Fld@bordercolor]
3178   }
3179   \tl_if_empty:NF\Fld@bcolor
3180   {
3181     /BG[\Fld@bcolor]
3182   }
3183   /CA(\Hy@escapestring{\Fld@cbsymbol})%
3184 >>
3185 /DA
3186 (
3187   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3188   \tl_if_empty:NF \Fld@color
3189   {
3190     \c_space_tl \Fld@color
3191   }
3192 )
3193 /H/P
3194 \legacy_if:nTF {Fld@checked}
3195 {
3196   /V/Yes /AS/Yes
3197 }
3198 {
3199   /V/Off /AS/Off
3200 }
3201 \Fld@additionalactions
3202 }
3203 \ExplSyntaxOff
3204 \ExplSyntaxOn
3205 \def\PDFForm@Push
3206 {
3207   /Subtype/Widget
3208   ~\Fld@annotflags
3209   ~\Fld@pageobjref
3210   ~\Fld@annotnames
3211   /FT/Btn
3212   ~\Fld@flags
3213   /H/P
3214   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3215   \bool_if:nT
3216   {
3217     !\int_compare_p:nNn {\Fld@rotation} = {0}
3218     ||
3219     \tl_if_exist_p:N \Fld@bordercolor
3220   }
3221   {
3222     /MK
3223     <<

```

```

3224         \int_compare:nNf {\Fld@rotation} = {0}
3225         {
3226             /R~\Fld@rotation
3227         }
3228         \tl_if_exist:NT \Fld@bordercolor
3229         {
3230             /BC[\Fld@bordercolor]
3231         }
3232         >>
3233     }
3234     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3235     \Fld@additionalactions
3236 }
3237
3238 \ExplSyntaxOff
3239 \def\PDFForm@List{%
3240     /Subtype/Widget%
3241     \Fld@annotflags
3242     \Fld@pageobjref
3243     \Fld@annotnames
3244     /FT/Ch%
3245     \Fld@flags
3246     /Q \Fld@align
3247     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3248     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3249         \ifx\Fld@bordercolor\relax\else 1\fi
3250         \ifx\Fld@bcolor\relax \else 1\fi
3251     \space
3252 \else
3253     /MK<<%
3254     \ifnum\Fld@rotation=\z@
3255     \else
3256         /R \Fld@rotation
3257     \fi
3258     \ifx\Fld@bordercolor\relax
3259     \else
3260         /BC[\Fld@bordercolor]%
3261     \fi
3262     \ifx\Fld@bcolor\relax
3263     \else
3264         /BG[\Fld@bcolor]%
3265     \fi
3266     >>%
3267 \fi
3268 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3269     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3270 \Fld@choices
3271 \Fld@additionalactions
3272 }
3273 \ExplSyntaxOn
3274 \def\PDFForm@Radio
3275 {
3276     /Subtype/Widget
3277     ~\Fld@annotflags

```

```

3278 ~\Fld@pageobjref
3279 ~\Fld@annotnames
3280 /FT/Btn
3281 \Fld@flags
3282 /H/P
3283 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3284 /MK<<
3285 \ifnum\Fld@rotation=\z@
3286 \else
3287 /R~\Fld@rotation
3288 \fi
3289 \ifx\Fld@bordercolor\relax
3290 \else
3291 /BC[\Fld@bordercolor]%
3292 \fi
3293 \ifx\Fld@bcolor\relax
3294 \else
3295 /BG[\Fld@bcolor]%
3296 \fi
3297 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3298 >>
3299 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3300 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3301 \ifx\Fld@default\@empty
3302 /V/Off%
3303 /DV/Off%
3304 \else
3305 /V/\Fld@default
3306 /DV/\Fld@default
3307 \fi
3308 \Fld@additionalactions
3309 }
3310 \ExplSyntaxOff
3311 \ExplSyntaxOn
3312 % Does an appearance dict make sense here?
3313 \def\PDFForm@Text
3314 {
3315 /Subtype/Widget
3316 ~\Fld@annotflags
3317 ~\Fld@pageobjref
3318 ~\Fld@annotnames
3319 /FT/Tx
3320 ~\Fld@flags
3321 /Q~\Fld@align
3322 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3323 \bool_if:nT
3324 {
3325 !\int_compare_p:nNn {\Fld@rotation} = {0}
3326 ||
3327 \tl_if_exist_p:N \Fld@bordercolor
3328 ||
3329 \tl_if_exist_p:N \Fld@bcolor
3330 }
3331 {

```

```

3332     /MK
3333     <<
3334     \int_compare:nNnF {\Fld@rotation} = {0}
3335     {
3336     /R~\Fld@rotation
3337     }
3338     \tl_if_exist:NT \Fld@bordercolor
3339     {
3340     /BC[\Fld@bordercolor]
3341     }
3342     \tl_if_exist:NT \Fld@bcolor
3343     {
3344     /BG[\Fld@bcolor]
3345     }
3346     >>
3347   }
3348 /DA
3349 (
3350   /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3351   \tl_if_empty:NF {\c_space_tl\Fld@color}
3352 )
3353 /DV(\Hy@escapestring{\Fld@default})
3354 /V(\Hy@escapestring{\Fld@value})
3355 ~\Fld@additionalactions
3356 \int_compare:nNnT { \Fld@maxlen}>{0}
3357 {
3358   /MaxLen~\Fld@maxlen
3359 }
3360 }
3361 \ExplSyntaxOff
3362
3363 \def\PDFForm@Submit{%
3364   /Subtype/Widget%
3365   \Fld@annotflags
3366   \Fld@pageobjref
3367   \Fld@annotnames
3368   /FT/Btn%
3369   \Fld@flags
3370   /H/P%
3371   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3372   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3373     \ifx\Fld@bordercolor\relax\else 1\fi
3374     \space
3375   \else
3376     /MK<<%
3377     \ifnum\Fld@rotation=\z@
3378     \else
3379     /R \Fld@rotation
3380     \fi
3381     \ifx\Fld@bordercolor\relax
3382     \else
3383     /BC[\Fld@bordercolor]%
3384     \fi
3385     >>%

```

```

3386 \fi
3387 /A<<%
3388 /S/SubmitForm%
3389 /F<<%
3390 /FS/URL%
3391 /F(\Hy@escapestring{\Form@action})%
3392 >>%
3393 \Fld@submitflags
3394 >>%
3395 \Fld@additionalactions
3396 }
3397 \ExplSyntaxOn
3398 \def\PDFForm@Reset{%
3399 /Subtype/Widget%
3400 \Fld@annotflags
3401 \Fld@pageobjref
3402 \Fld@annotnames
3403 /FT/Btn%
3404 \Fld@flags
3405 /H/P%
3406 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3407 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3408 \ifx\Fld@bordercolor\relax\else 1\fi
3409 \space
3410 \else
3411 /MK<<%
3412 \ifnum\Fld@rotation=\z@
3413 \else
3414 /R~\Fld@rotation
3415 \fi
3416 \ifx\Fld@bordercolor\relax
3417 \else
3418 /BC[\Fld@bordercolor]%
3419 \fi
3420 >>%
3421 \fi
3422 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3423 /A<</S/ResetForm>>%
3424 \Fld@additionalactions
3425 }%
3426
3427
3428 %these patterns are used in hyperref checks.
3429 %it is unclear if they are really useful and if a backend support is
3430 %needed.
3431 \str_case:VnF \c_sys_backend_str
3432 {
3433 { pdfmode }
3434 {
3435 \def\HyPat@ObjRef
3436 {
3437 [0-9]*[1-9][0-9]*~0~R
3438 }
3439 }

```



```

3440 { dvipdfmx }
3441 {
3442   \def\HyPat@ObjRef
3443     {
3444       @[\~~]+
3445     }
3446 }
3447 { xdvipdfmx }
3448 {
3449   \def\HyPat@ObjRef
3450     {
3451       @[\~~]+
3452     }
3453 }
3454 }
3455 { %also set in hyperref sty, so probably not needed.
3456   \def\HyPat@ObjRef/{.+}
3457 }
3458
3459
3460 \ExplSyntaxOff
3461 % UF: removed Hy@writebookmark
3462 %   \Hy@currentbookmarklevel{0}
3463 %   \Hy@numberline
3464 %   \__hypwritetorep
3465 %   counter{bookmark@seq@number}
3466 % removed \HyPsd@SanitizeForOutFile, not needed
3467 % removed \currentpdfbookmark, defined by bookmark,
3468 % should use \newcommand there
3469 % removed \subpdfbookmark, defined by bookmark,
3470 % should use \newcommand there
3471 % removed \belowpdfbookmark, defined by bookmark,
3472 % should use \newcommand there
3473 % removed \pdfbookmark, defined by bookmark,
3474 % \BOOKMARK
3475 % \@BOOKMARK
3476 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3477 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3478 %% removed \ReadBookmarks / unneeded with bookmark.
3479 %% removed \Hy@OutlineName
3480 %% removed \check@bm@number
3481 %% removed \calc@bm@number
3482
3483 \ifHy@implicit
3484 \else
3485   \expandafter\endinput
3486 \fi
3487 \newlength\Hy@SectionHShift
3488 \def\Hy@SectionAnchorHref#1{%
3489   \ifx\protect\@typeset@protect
3490     \Hy__hypSectionAnchor{#1}%
3491   \fi
3492 }
3493 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%

```

```

3494 \leavevmode
3495 \hbox to Opt{%
3496   \kern-\Hy@SectionHShift
3497   \Hy@raisedlink{%
3498     \hyper@anchorstart{#1}\hyper@anchorend
3499   }%
3500   \hss
3501 }%
3502 }
3503 \let\H@old@ssect\@ssect
3504 \def\@ssect#1#2#3#4#5{%
3505   \Hy@MakeCurrentHrefAuto{section*}%
3506   \setlength{\Hy@SectionHShift}{#1}%
3507   \begingroup
3508     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3509     \toks\tw@expandafter{%
3510       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3511       #5%
3512     }%
3513   \edef\x{\endgroup
3514     \the\toks@\the\toks\tw@}%
3515   }\x
3516 }
3517 \let\H@old@schapter\@schapter
3518 \def\@schapter#1{%
3519   \begingroup
3520     \let\@mkboth\@gobbletwo
3521     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3522     \Hy@raisedlink{%
3523       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3524     }%
3525   \endgroup
3526   \H@old@schapter{#1}%
3527 }
3528 \ltx@ifundefined{chapter}{-}{%
3529   \let\Hy@org@chapter\@chapter
3530   \def\@chapter{%
3531     \def\Hy@next{%
3532       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3533       \Hy@raisedlink{%
3534         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3535       }%
3536     }%
3537     \ifnum\c@secnumdepth>\m@ne
3538       \ltx@ifundefined{if@mainmatter}%
3539       \iftrue{\csname if@mainmatter\endcsname}%
3540       \let\Hy@next\relax
3541     \fi
3542     \fi
3543     \Hy@next
3544     \Hy@org@chapter
3545   }%
3546 }
3547 \let\H@old@part\@part

```

```

3548 \begingroup\expandafter\expandafter\expandafter\endgroup
3549 \expandafter\ifx\csname chapter\endcsname\relax
3550   \let\Hy@secnum@part\z@
3551 \else
3552   \let\Hy@secnum@part\m@ne
3553 \fi
3554 \def\@part{%
3555   \ifnum\Hy@secnum@part>\c@secnumdepth
3556     \phantomsection
3557   \fi
3558   \H@old@part
3559 }
3560 \let\H@old@spart\@spart
3561 \def\@spart#1{%
3562   \Hy@MakeCurrentHrefAuto{part*}%
3563   \Hy@raisedlink{%
3564     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3565   }%
3566   \H@old@spart{#1}%
3567 }
3568 \let\H@old@sect\@sect
3569 \def\@sect#1#2#3#4#5#6[#7]#8{%
3570   \ifnum #2>\c@secnumdepth
3571     \expandafter\@firstoftwo
3572   \else
3573     \expandafter\@secondoftwo
3574   \fi
3575   {%
3576     \Hy@MakeCurrentHrefAuto{section*}%
3577     \setlength{\Hy@SectionHShift}{#3}%
3578     \begingroup
3579     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3580     \toks\tw@\expandafter{%
3581       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3582       #8%
3583     }%
3584     \edef\x{\endgroup
3585       \the\toks@\the\toks\tw@}%
3586     }\x
3587   }{%
3588     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3589   }%
3590 }
3591 \expandafter\def\csname Parent-4\endcsname{}
3592 \expandafter\def\csname Parent-3\endcsname{}
3593 \expandafter\def\csname Parent-2\endcsname{}
3594 \expandafter\def\csname Parent-1\endcsname{}
3595 \expandafter\def\csname Parent0\endcsname{}
3596 \expandafter\def\csname Parent1\endcsname{}
3597 \expandafter\def\csname Parent2\endcsname{}
3598 \expandafter\def\csname Parent3\endcsname{}
3599 \expandafter\def\csname Parent4\endcsname{}
3600 %%
3601 %% End of file 'hgeneric-testphase.def'.

```

```

3602 </package>
3603 <*colorscheme>
3604 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3605 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3606 % linkcolor=
3607 %,filecolor=
3608 %,urlcolor=
3609 %,menucolor=
3610 %,runcolor=
3611 %,linkbordercolor=
3612 %,filebordercolor=
3613 %,urlbordercolor=
3614 %,menubordercolor=
3615 %,runbordercolor=
3616
3617 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3618 {
3619   linkcolor      = [rgb]{1,0,0}, %red
3620   filecolor      = [rgb]{0,1,1}, %cyan
3621   urlcolor       = [rgb]{1,0,1}, %magenta
3622   menucolor      = [rgb]{1, 0, 0}, %red
3623   runcolor       = [rgb]{0,1,1}, %cyan
3624 %-----
3625   linkbordercolor = [rgb]{1, 0 ,0 },
3626   filebordercolor = [rgb]{0, .5, .5},
3627   urlbordercolor  = [rgb]{0, 1, 1},
3628   menubordercolor = [rgb]{1, 0, 0},
3629   runbordercolor  = [rgb]{0, .7, .7}
3630 }
3631
3632 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3633 {
3634   linkcolor      = [rgb]{0,0.2,0.6},
3635   filecolor      = [rgb]{0.8,0,0.8},
3636   urlcolor       = [rgb]{0.8,0,0.8},
3637   menucolor      = [rgb]{0,0.2,0.6},
3638   runcolor       = [rgb]{0.8,0,0.8},
3639 %----- %-----
3640   linkbordercolor = [rgb]{0,0.2,0.6},
3641   filebordercolor = [rgb]{0.8,0,0.8},
3642   urlbordercolor  = [rgb]{0.8,0,0.8},
3643   menubordercolor = [rgb]{0,0.2,0.6},
3644   runbordercolor  = [rgb]{0.8,0,0.8}
3645 }
3646
3647 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3648 { %two colors: intern/extern
3649   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3650   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3651   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3652   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3653   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3654 %----- %-----
3655   linkbordercolor = [rgb]{0.79216, 0, 0.12549},

```

```

3656   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3657   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3658   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3659   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3660 }
3661
3662 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3663 { %all darkgray
3664   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3665   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3666   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3667   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3668   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3669   %----- %-----
3670   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3671   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3672   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3673   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3674   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3675 }
3676
3677 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3678 { %dvipsnam.def
3679   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3680   filecolor      = [rgb]{1, 0, 0}, %Red
3681   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3682   menucolor      = [rgb]{1, 0, 0}, %Red
3683   runcolor       = [rgb]{1, 0, 0}, %Red
3684   %----- %-----
3685   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3686   filebordercolor = [rgb]{1, 0, 0}, %Red
3687   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3688   menubordercolor = [rgb]{1, 0, 0}, %Red
3689   runbordercolor  = [rgb]{1, 0, 0} %Red
3690 }
3691
3692 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3693 { %dvipsnam.def
3694   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3695   filecolor      = [rgb]{0, 1, 0}, %Green
3696   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3697   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3698   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3699   %----- %-----
3700   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3701   filebordercolor = [rgb]{0, 1, 0}, %Green
3702   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3703   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3704   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3705 }
3706
3707
3708 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3709 {

```

```

3710 linkcolor      = [rgb]{0.50196, 0, 0.02353},
3711 filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3712 urlcolor       = [rgb]{0.54118, 0, 0.52941},
3713 menucolor      = [rgb]{0.44706, 0.45882, 0},
3714 runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3715 %-----      %-----
3716 linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3717 filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3718 urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3719 menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3720 runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3721 }
3722
3723 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3724 {
3725 linkcolor      = [rgb]{0,0,0},
3726 filecolor      = [rgb]{0,0,0},
3727 urlcolor       = [rgb]{0,0,0},
3728 menucolor      = [rgb]{0,0,0},
3729 runcolor       = [rgb]{0,0,0},
3730 %-----      %-----
3731 linkbordercolor = [rgb]{0,0,0},
3732 filebordercolor = [rgb]{0,0,0},
3733 urlbordercolor  = [rgb]{0,0,0},
3734 menubordercolor = [rgb]{0,0,0},
3735 runbordercolor  = [rgb]{0,0,0}
3736 }
3737 </colorscheme>

```

# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

<b>Symbols</b>	
\#	265, 723
\\$	264
\%	724
\.	533, 537, 539
@curropt commands:	
\@curropt:	2647, 2714
\\	17, 18, 19, 26, 30, 40, 60, 67, 74, 81, 88, 103, 108, 109, 117, 118, 127, 128, 129, 130, 131, 138, 145, 266, 267, 853, 2649, 2716, 3056
\_	531, 533, 537, 539
<b>A</b>	
\A	531
\Acrobatmenu	18, 158
\addcontentsline	13
\AddToDocumentProperties	378
\AddToHook	416, 429
\AddToHookNext	183
\advance	2580, 2654, 2698, 2699
allcolors (hypersetup key)	966
\author	2
<b>B</b>	
\b	535
\begingroup	194, 263, 365, 2486, 2614, 2643, 2655, 2689, 3078, 3507, 3519, 3548, 3578
\belowpdfbookmark	3471
\bgroup	260, 365
\BOOKMARK	3474
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	253, 283, 345, 670, 686, 698, 711, 719, 749, 807, 864, 936, 946, 1190, 1201
\bool_if:nTF	819, 1897, 1922, 3215, 3323
\bool_lazy_and:nnTF	413
\bool_lazy_or:nnTF	1220, 1269
\bool_new:N	188, 189, 517, 521, 525
\bool_set_true:N	526
bordercolormodel (hypersetup key)	13, <u>987</u>
box commands:	
\box_dp:N	607, 2799, 2850, 2885
\box_ht:N	606, 2798, 2849, 2884
\box_new:N	466, 528
\box_set_dp:Nn	2094
\box_set_ht:Nn	2101
\box_use:N	1207, 2801, 2859, 2888
\box_use_drop:N	1212
\box_wd:N	605, 2797, 2848, 2883
\l_tmpa_box	2795, 2797, 2798, 2799, 2801, 2834, 2848, 2849, 2850, 2859, 2881, 2883, 2884, 2885, 2888
<b>C</b>	
\catcode	264, 265
\char	2733, 2930, 2942
\chardef	143
\cite	32
clist commands:	
\clist_map_function:nN	110, 119
\clist_map_inline:nn	2273
\clist_map_variable:nNn	2646, 2713
color commands:	
\color_export:nnN	38, 421, 895, 1008
\color_select:n	903, 939, 1211, 2840, 2844
\color_select:nn	909
\color_set:nn	2, 383, 920
\color_set:nnn	2, 382, 926
color names:	
hyp/annot/file	514
hyp/annot/link	514
hyp/annot/menu	514
hyp/annot/run	514
hyp/annot/url	514
colorfile (hypersetup key)	966
colorlink (hypersetup key)	966
colorlinks (hypersetup key)	952
colormenu (hypersetup key)	966
colorryn (hypersetup key)	966
colorscheme (hypersetup key)	1, <u>1378</u>
colorurl (hypersetup key)	966
cs commands:	
\cs_generate_variant:Nn	141, 142, 380, 445, 460, 465, 569, 895, 912, 929
\cs_gset:Npn	1182
\cs_gset_eq:NN	399, 410
\cs_if_exist_p:N	414
\cs_new:Npn	103, 461
\cs_new_protected:Npn	194, 292, 310, 328, 355, 375, 390, 401, 439, 448, 545, 549, 553, 557, 570, 572, 576, 620, 628, 637, 641, 663,

668, 684, 696, 709, 747, 805, 862, 896, 907, 913, 924, 1126, 2129, 2200	draft (hypersetup key) . . . . .	<u>1402</u>
<code>\cs_set:Nn</code> . . . . .	<b>E</b>	
<code>\cs_set_eq:NN</code> . . . . .	<code>\edef</code> 2488, 2657, 2693, 2700, 3041, 3513, 3584	
. . . . . 146, 151, 152, 723, 724, 1455	<code>\EdefEscapeName</code> . . . . .	2712, 2717
<code>\cs_set_protected:Npn</code> . . . . .	<code>\egroup</code> . . . . .	368
. . . . . 144, 165, 586, 853	<code>\else</code> . . . . .	2485, 2610, 2664, 2668, 2671, 2688, 2696, 3029, 3034, 3051, 3057, 3069, 3073, 3090, 3094, 3098, 3102, 3106, 3110, 3114, 3118, 3122, 3126, 3143, 3147, 3248, 3249, 3250, 3252, 3255, 3259, 3263, 3269, 3286, 3290, 3294, 3300, 3304, 3372, 3373, 3375, 3378, 3382, 3407, 3408, 3410, 3413, 3417, 3484, 3551, 3572
<code>\csname</code> . . . . .	<code>\endcsname</code> . . . . .	2675, 3025, 3048, 3059, 3539, 3549, 3591, 3592, 3593, 3594, 3595, 3596, 3597, 3598, 3599
. . . . . 2661, 3048, 3059, 3539, 3549, 3591, 3592, 3593, 3594, 3595, 3596, 3597, 3598, 3599	<code>\endgroup</code> . . . . .	269, 272, 366, 368, 2489, 2634, 2657, 2703, 2706, 3085, 3513, 3525, 3548, 3584
<code>\currentpdfbookmark</code> . . . . .	<code>\endinput</code> . . . . .	3485
	exp commands:	
<b>D</b>		
<code>\d</code> . . . . .	<code>\exp_args:Ne</code> . . . . .	3133
debug (hypersetup key) . . . . .	<code>\exp_args:Nne</code> . . . . .	2238
<u>1402</u>	<code>\exp_args:Nno</code> . . . . .	312, 330
<code>\DeclareDocumentMetadata</code> . . . . .	<code>\exp_args:Nno</code> . . . . .	294, 357, 2186
. . . . . 3, 10, 14, 15, 18, 19, 27, 33	<code>\exp_args:Nnox</code> . . . . .	757
<code>\DeclareRobustCommand</code> . . . . .	<code>\exp_args:NNV</code> . . . . .	1882, 1913
. . . . . 247, 276, 300, 318, 338, 368, 3493	<code>\exp_args:NnV</code> . . . . .	622, 631
<code>\def</code> . . . . .	<code>\exp_args:Nnx</code> . . . . .	2326, 2431
. . . . . 150, 365, 366, 368, 581, 2449, 2453, 2460, 2478, 2507, 2520, 2578, 2587, 2602, 2603, 2606, 2607, 2620, 2621, 2637, 2638, 2641, 2642, 2680, 2709, 2761, 2775, 2776, 2788, 2816, 2818, 2819, 2829, 2864, 2866, 2867, 2878, 2897, 2899, 2900, 2902, 2903, 2912, 2959, 3026, 3036, 3038, 3043, 3055, 3066, 3077, 3088, 3131, 3140, 3152, 3205, 3239, 3274, 3313, 3363, 3398, 3435, 3442, 3449, 3456, 3488, 3504, 3518, 3530, 3531, 3554, 3561, 3569, 3591, 3592, 3593, 3594, 3595, 3596, 3597, 3598, 3599	<code>\exp_args:No</code> . . . . .	. . . . . 655, 659, 2778, 2821, 2869, 2904
<code>\DefaultHeightofCheckBox</code> . . . . .	<code>\exp_args:NV</code> . . . . .	603, 610
2903	<code>\exp_args:Nx</code> . . . . .	753, 2179, 2227, 2480
<code>\DefaultHeightofChoiceMenu</code> . . . . .	<code>\exp_not:N</code> . . . . .	853
2642	<code>\exp_not:n</code> . . . . .	1317, 1351, 1462, 1522, 1543, 1580, 1600, 1623, 1667, 1700, 1725, 1776, 1801, 1828, 1877, 1954, 1982, 2009, 2029, 2259, 2376, 2404, 2414, 2425
<code>\DefaultHeightofReset</code> . . . . .	<code>\expandafter</code> . . . . .	266, 2615, 2649, 2658, 2660, 2661, 2716, 3042, 3058, 3485, 3509, 3510, 3548, 3549, 3571, 3573, 3580, 3581, 3591, 3592, 3593, 3594, 3595, 3596, 3597, 3598, 3599
2867	<code>\ExplSyntaxOff</code> . . . . .	. . . . . 9, 2517, 2570, 2597, 2955, 3023, 3139, 3203, 3238, 3310, 3361, 3460
<code>\DefaultHeightofSubmit</code> . . . . .	<code>\ExplSyntaxOn</code> . . . . .	. . . . . 7, 10, 2518, 2575, 2598, 2958, 3130, 3151, 3204, 3273, 3311, 3397
2819	<code>\extension</code> (hypersetup key) . . . . .	<u>13</u> , <u>1424</u>
<code>\DefaultHeightofText</code> . . . . .		
2611		
<code>\DefaultHeightofTextMultiline</code> . . . . .		
2609		
<code>\DefaultOptionsofCheckBox</code> . . . . .		
2906		
<code>\DefaultOptionsofPushButton</code> . . . . .		
2780		
<code>\DefaultOptionsofReset</code> . . . . .		
2871		
<code>\DefaultOptionsofSubmit</code> . . . . .		
2823		
<code>\DefaultOptionsofText</code> . . . . .		
2616		
<code>\DefaultWidthofCheckBox</code> . . . . .		
2902		
<code>\DefaultWidthofChoiceMenu</code> . . . . .		
2641		
<code>\DefaultWidthofReset</code> . . . . .		
2866		
<code>\DefaultWidthofSubmit</code> . . . . .		
2818		
<code>\DefaultWidthofText</code> . . . . .		
2606		
destlabel (hypersetup key) . . . . .		
<u>13</u>		
dim commands:		
<code>\dim_to_decimal_in_bp:n</code> . . . . .		
. . . . . 11, 146		
<code>\do</code> . . . . .		
2647, 2714		
<code>\documentclass</code> . . . . .		
20		



<b>F</b>	
<code>\fbox</code> .....	2752, 2840, 2844
<code>\fi</code> .....	2474,
	2501, 2504, 2513, 2612, 2620, 2621,
	2652, 2670, 2673, 2674, 2680, 2683,
	2692, 2694, 2701, 2704, 3031, 3053,
	3063, 3065, 3071, 3075, 3082, 3083,
	3092, 3096, 3100, 3104, 3108, 3112,
	3116, 3120, 3124, 3128, 3145, 3149,
	3248, 3249, 3250, 3257, 3261, 3265,
	3267, 3269, 3288, 3292, 3296, 3300,
	3307, 3372, 3373, 3380, 3384, 3386,
	3407, 3408, 3415, 3419, 3421, 3486,
	3491, 3541, 3542, 3553, 3557, 3574
file (hypersetup key) .....	10, 1470
file commands:	
<code>\file_input:n</code> .....	8
fileborderstyle (hypersetup key) .....	14, 1080
filecolor (hypersetup key) .....	966
final (hypersetup key) .....	1402
<code>\fontencoding</code> .....	2729, 2926, 2938
<code>\fontfamily</code> .....	2728, 2925, 2937
<code>\fontseries</code> .....	2730, 2927, 2939
<code>\fontshape</code> .....	2731, 2928, 2940
fp commands:	
<code>\fp_eval:n</code> .....	2052, 2080, 2096
<b>G</b>	
<code>\gdef</code> .....	266, 267
<code>\GetDocumentProperties</code> .....	2
<code>\global</code> .....	2580, 3021
group commands:	
<code>\group_begin:</code> ..	250, 280, 304, 322,
	342, 559, 673, 689, 713, 751, 809,
	866, 938, 1193, 1210, 2355, 2727,
	2777, 2820, 2868, 2901, 2924, 2936
<code>\group_end:</code> .....	295, 313, 331,
	360, 566, 678, 701, 737, 797, 847,
	890, 948, 1213, 1216, 2362, 2734,
	2813, 2861, 2894, 2931, 2943, 2953
<b>H</b>	
<code>\hbox</code> .....	365, 3495
hbox commands:	
<code>\hbox_overlap_right:n</code> .....	1207
<code>\hbox_set:Nn</code> .....	2795, 2834, 2881
<code>\hbox_set:Nw</code> .....	1194
<code>\hbox_set_end:</code> .....	1203
<code>\hbox_set_to_wd:Nnn</code> .....	2078
hidefile (hypersetup key) .....	1354
hidelink (hypersetup key) .....	1354
hidelinks (hypersetup key) .....	1354
hidemenu (hypersetup key) .....	1354
hiderun (hypersetup key) .....	1354
hideurl (hypersetup key) .....	1354
hook commands:	
<code>\hook_gput_code:nnn</code> .....	
	..... 392, 403, 932, 942, 1186, 1197
<code>\hook_new:n</code> .....	381, 544, 619
<code>\hook_new_pair:nn</code> .....	
	..... 245, 274, 299, 317, 336
<code>\hook_use:n</code> .....	
	. 249, 270, 279, 296, 303, 314, 321,
	332, 341, 361, 561, 625, 634, 675, 691
<code>\href</code> .....	3–5, 19, 247
<code>\hreflaunch</code> .....	5
<code>\hrefpdf</code> .....	4, 5, 19, 300, 373
<code>\hrefrun</code> .....	4, 318, 374
<code>\hrefurl</code> .....	4, 5, 19, 276, 372
<code>\hss</code> .....	3500
Hy internal commands:	
<code>\Hy_hyescapeform</code> ..	3042, 3055, 3061
<code>\Hy_hypSectionAnchor</code> ..	3490, 3493
HyField internal commands:	
<code>\HyField_hypAddToFields</code> ..	2480, 2507
hyp commands:	
<code>\l_hyp_annot_colorfile_bool</code> ...	515
<code>\l_hyp_annot_colorlink_bool</code> ...	515
<code>\l_hyp_annot_colormenu_bool</code> ...	515
<code>\l_hyp_annot_colorrund_bool</code> ....	515
<code>\l_hyp_annot_colorurl_bool</code> ....	515
<code>\l_hyp_annot_ocgcolorfile_bool</code> .	519
<code>\l_hyp_annot_ocgcolorlink_bool</code> .	519
<code>\l_hyp_annot_ocgcolormenu_bool</code> .	519
<code>\l_hyp_annot_ocgcolorrund_bool</code> ..	519
<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	519
<code>\l_hyp_current_dest_name_tl</code> ....	
	..... 618, 674, 690
hyp internal commands:	
<code>\g__hyp_AcroForm_CoFields_prop</code> ..	
	..... 2441, 2462, 2531, 2533, 2548
<code>\g__hyp_AcroForm_Fields_prop</code> ...	
	..... 2442, 2455, 2526
<code>\l__hyp_annot_GoTo_bool</code> 670, 686, 698	
<code>\l__hyp_annot_GoTo_bool\l__-</code>	
<code>hyp_annot_URI_bool\l__-</code>	
<code>hyp_annot_GoToR_bool\l__-</code>	
<code>hyp_annot_Named_bool\l__-</code>	
<code>hyp_annot_Launch_bool</code> .....	523
<code>\l__hyp_annot_GoToR_bool</code> .....	749
<code>\l__hyp_annot_Launch_bool</code> .....	807
<code>\l__hyp_annot_Named_bool</code> .....	864
<code>\c__hyp_annot_types_seq</code> .....	
	..... 478, 515, 519, 966, 1267, 1365
<code>\l__hyp_annot_URI_bool</code> .....	711
<code>\g__hyp_bordercolormodel_str</code> ...	
	..... 423, 514, 991, 1010

\_hyp_citebordercolor_hook_- init: . . . . .	388, 401, 410	\l_hyp_href_url_encode_bool . . . . . . . .	188, 199, 237, 253, 283, 345
\_hyp_citecolor_hook_init: . . . . .	386, 390, 399	\_hyp_href_url_format: . . . . .	194, 200, 242, 358
\_hyp_clist_display:n . . . . .	103, 110, 119	\l_hyp_href_url_ismap_bool . . . . .	189, 213, 719
\_hyp_color_select:n <a href="#">38</a> , <a href="#">896</a> , 896, 912		\l_hyp_href_url_protocol_tl . . . . .	190, 201, 240, 294, 359
\_hyp_color_select_aux:wn . . . . .	<a href="#">896</a> , 900, 907	\_hyp_link_goto_begin:nw . . . . .	641, 676, 692
\_hyp_color_set:nn . . . . .	<a href="#">38</a> , <a href="#">39</a> , 386, 388, 420, <a href="#">913</a> , 913, 929, 971, 1007	\_hyp_link_goto_end: . . . . .	663, 677, 700
\_hyp_color_set_aux:nwn <a href="#">913</a> , 917, 924		\c_hyp_map_annot_hyp_prop . . . . .	<a href="#">478</a>
\c_hyp_colorscheme_daleif_prop <a href="#">3632</a>		\c_hyp_map_hyp_annot_prop . . . . .	<a href="#">478</a> , 930, 994, 1034, 1061, 1069, 1080, 1107, 1115, 1184, 1292, 1328, 1338, 1470
\c_hyp_colorscheme_henryford_- prop . . . . .	3723	\_hyp_ocg_init: <a href="#">1126</a> , 1126, 1182, 1192	
\c_hyp_colorscheme_julian_prop <a href="#">3647</a>		\l_hyp_page/Trans . . . . .	<a href="#">542</a>
\c_hyp_colorscheme_phelype_prop . . . . .	<a href="#">3708</a>	\_hyp_PageLabels_gpush: <a href="#">576</a> , <a href="#">576</a> , <a href="#">589</a>	
\c_hyp_colorscheme_szabolcsA_- prop . . . . .	<a href="#">3677</a>	\l_hyp_para_tmpa_tl <a href="#">474</a> , 818, 821, 830	
\c_hyp_colorscheme_szabolcsB_- prop . . . . .	<a href="#">3692</a>	\l_hyp_para_tmpa_tl <sub>uuu</sub> \l_hyp_- text_tmpa_str <sub>uuu</sub> \g_hyp_text_- tmpa_str . . . . .	<a href="#">471</a>
\c_hyp_colorscheme_tivv_prop . . . . .	<a href="#">3662</a>	\_hyp_ref_check:nn <a href="#">438</a> , 448, 460, 2585	
\l_hyp_dest_box . . . . .	<a href="#">31</a> , <a href="#">528</a> , 605, 606, 607, 2078, 2094, 2101	\_hyp_ref_if_exist:nn . . . . .	<a href="#">438</a> , 446, 447
\l_hyp_dest_name_tmpa_tl . . . . .	<a href="#">471</a> , 644, 645, 651, 655, 657, 660, 770, 783	\_hyp_ref_if_exist:nnTF . . . . .	450, 2589
\l_hyp_dest_pdfremotestartview_- tl . . . . .	<a href="#">497</a> , 778, 1885, 1889	\_hyp_ref_label:nn <a href="#">438</a> , 439, 445, 2584	
\g_hyp_dest_pdfstartpage_tl . . . . .	<a href="#">497</a> , 1896, 1898, 1905, 1923, 1930	\_hyp_ref_value:nn <a href="#">438</a> , 461, 465, 2593	
\g_hyp_dest_pdfstartview_tl . . . . .	<a href="#">497</a> , 1898, 1905, 1916, 1920, 1923, 1930	\_hyp_secondoftwoithopt:wnn . . . . .	371, 372, 373, 374
\l_hyp_dest_pdfview_tl . . . . .	<a href="#">513</a> , 623, 632, 2047, 2050, 2057, 2060, 2061, 2062, 2063, 2064, 2065, 2070, 2074, 2110	\_hyp_setup_info_date_key:nn . . . . .	2200, 2232, 2233
\c_hyp_dest_startview_regex . . . . .	<a href="#">529</a> , 1883, 1914	\_hyp_setup_info_key:nn . . . . .	2129, 2193, 2194, 2195, 2196, 2198, 2199
\c_hyp_dest_undefined_tl . . . . .	<a href="#">477</a> , 650, 651	\_hyp_store_metadata:nn . . . . .	375, 380, 2126, 2161, 2179, 2213, 2227, 2248, 2320
\_hyp_destination:nn . . . . .	<a href="#">31</a> , <a href="#">592</a> , 592, 623, 632	\_hyp_text_cleanup:N . . . . .	<a href="#">549</a> , 549, 563
\l_hyp_filename_tmpa_tl . . . . .	<a href="#">471</a> , 752, 754, 760, 761, 766, 813, 814, 830	\l_hyp_text_enc_dest_print_tl . . . . .	<a href="#">500</a> , 769
\_hyp_href_pdf_aux:nn . . . . .	307, 310	\l_hyp_text_enc_dest_tl . . . . .	<a href="#">31</a> , <a href="#">500</a> , 599, 656
\l_hyp_href_pdf_destination_tl . . . . .	<a href="#">191</a> , 202, 241, <a href="#">312</a>	\l_hyp_text_enc_file_print_tl . . . . .	<a href="#">500</a> , 812
\l_hyp_href_pdf_page_tl <a href="#">192</a> , 210, <a href="#">777</a>		\l_hyp_text_enc_info_print_tl . . . . .	<a href="#">500</a> , 574
\_hyp_href_run_aux:nn . . . . .	<a href="#">325</a> , <a href="#">328</a>	\l_hyp_text_enc_para_print_tl . . . . .	<a href="#">500</a> , 817
\l_hyp_href_run_parameter_tl . . . . .	<a href="#">193</a> , 214, <a href="#">330</a>	\l_hyp_text_enc_uri_print_tl . . . . .	255, 258, 285, 288, 347, 350, <a href="#">500</a> , 716, 1481
\_hyp_href_url_aux:n . . . . .	<a href="#">352</a> , <a href="#">355</a>	\_hyp_text_pdfstring:nnN . . . . .	<a href="#">557</a> , 557, 569, 571, 574, 597, 654, 714, 767, 810, 815, 1481
\_hyp_href_url_aux:nn . . . . .	290, 292		

\_hyp\_text\_pdfstring\_info:nN . . .	1354
. . . . . 572, 572, 2155, 2173, 2183	1354
\_hyp\_text\_purify:nN . . . 545, 545, 562	1424
\_hyp\_text\_string\_from\_unicode:nN	10, 1470
. . . . . 553, 553, 564	13
\g\_hyp\_text\_tmpa\_str . . . 476, 565, 567	14, 1080
\l\_hyp\_text\_tmpa\_str . . . . .	966
. . . . . 475, 562, 563, 564, 565	1424
\l\_hyp\_tmpa\_box 466, 1194, 1207, 1212	1424
\l\_hyp\_tmpa\_int . . . . . 466	1424
\l\_hyp\_tmpa\_seq . . . . . 466, 1883,	1424
1885, 1914, 1916, 2037, 2038, 2042,	10, 1470
2044, 2068, 2075, 2076, 2077, 2086,	13
2088, 2098, 2103, 2535, 2537, 2543	14, 1080
\l\_hyp\_tmpa\_str . . . . .	966
. . . . . 466, 2155, 2156, 2158,	1424
2173, 2174, 2176, 2183, 2184, 2188	13
\l\_hyp\_tmpa\_tl 466, 600, 604, 612,	1220
1011, 1015, 1481, 1482, 1487, 1881,	1220
1883, 1912, 1914, 2044, 2045, 2052	1220
\l\_hyp\_uri\_tmpa\_tl . . . 471, 717, 718	1220
hyp/anchor . . . . . 618	1220
hyp/annot/file (color name) . . . . . 514	1220
hyp/annot/link (color name) . . . . . 514	1424
hyp/annot/menu (color name) . . . . . 514	2129
hyp/annot/run (color name) . . . . . 514	13
hyp/annot/url (color name) . . . . . 514	14, 1080
hyp/text/pdfstring . . . . . 544	14, 2200
\hypercalcbp . . . . . 11, 17, 146	2129
\HyperDestNameFilter . . . . . 13, 598, 655	1392
\hypersetup 1, 2, 5, 8, 10, 13, 18, 66, 99, 165	2262
\hypersetup keys:	2129
allcolors . . . . . 966	pdfkeywords . . . . . 2129
bookmarkstype . . . . . 13	pdflang . . . . . 14, 2115
bordercolormodel . . . . . 13, 987	pdflinkmargin . . . . . 14
colorfile . . . . . 966	pdfmoddate . . . . . 14, 2200
colorlink . . . . . 966	pdfproducer . . . . . 2129
colorlinks . . . . . 952	pdfremotestartview . . . . . 11
colormenu . . . . . 966	pdfstartview . . . . . 11
colorrund . . . . . 966	pdfsubject . . . . . 2129
colorscheme . . . . . 1, 1378	pdftitle . . . . . 2129
colorurl . . . . . 966	pdftrapped . . . . . 2234
debug . . . . . 1402	pdfversion . . . . . 1392
destlabel . . . . . 13	pdfview . . . . . 11, 2033
draft . . . . . 1402	plainpages . . . . . 1424
extension . . . . . 13, 1424	run . . . . . 10, 1470
file . . . . . 10, 1470	runborder . . . . . 13
fileborderstyle . . . . . 14, 1080	runborderstyle . . . . . 14, 1080
filecolor . . . . . 966	runcolor . . . . . 966
final . . . . . 1402	unicode . . . . . 1392
hidefile . . . . . 1354	url . . . . . 10, 1470
hidelink . . . . . 1354	urlborder . . . . . 13
hidelinks . . . . . 1354	urlborderstyle . . . . . 14, 1080
hidemenu . . . . . 1354	urlcolor . . . . . 966
	verbose . . . . . 1402

hypertextnames (hypersetup key) . . . .	<a href="#">1424</a>	\LayoutChoiceField . . . . .	2684
hypListbox internal commands:		\LayoutPushButtonField . . . . .	2790, 2807
\_hypListbox . . . . .	2702, 2761	\LayoutTextField . . . . .	2622
hypRadio internal commands:		\leavevmode . . . . .	2623, 2720, 3494
\_hypRadio . . . . .	2687, 2709	legacy commands:	
hypwritetorep internal commands:		\_legacy_if:nTF . . . . .	153, 427, 434, 584, 2560, 2786, 2827, 2877, 2910, 3194
\_hypwritetorep . . . . .	3464	\let . . . . .	252, 282, 306, 324, 344, 372, 373, 374, 681, 732, 739, 793, 799, 843, 849, 892, 2444, 2445, 2447, 2448, 2475, 2571, 2572, 2573, 2604, 2605, 2639, 2640, 2682, 3021, 3028, 3030, 3035, 3052, 3086, 3503, 3517, 3520, 3529, 3540, 3547, 3550, 3552, 3560, 3568
<b>I</b>			
\ifcase . . . . .	3248, 3372, 3407	link (hypersetup key) . . . . .	10, 1470
\ifcsname . . . . .	3025	linkborder (hypersetup key) . . . . .	13
\ifdim . . . . .	2652, 2691, 2692	linkborderstyle (hypersetup key) . . . . .	14, 1080
\ifnum . . . . .	3079, 3248, 3254, 3285, 3372, 3377, 3407, 3412, 3537, 3555, 3570	linkcolor (hypersetup key) . . . . .	966
\iftrue . . . . .	3539	linkfileprefix (hypersetup key) . . . . .	1424
\ifx . . . . .	2484, 2621, 2681, 3056, 3068, 3072, 3089, 3093, 3097, 3101, 3105, 3109, 3113, 3117, 3121, 3125, 3142, 3146, 3249, 3250, 3258, 3262, 3269, 3289, 3293, 3300, 3301, 3373, 3381, 3408, 3416, 3489, 3549	linktoc (hypersetup key) . . . . .	1424
\immediate . . . . .	2468, 2471	linktocpage (hypersetup key) . . . . .	1424
int commands:		localanchorname (hypersetup key) . . . . .	1424
\_int_compare:nNnTF . . . . .	2042, 2068, 2753, 3171, 3224, 3334, 3356	<b>M</b>	
\_int_compare_p:nNn . . . . .	3217, 3325	\MakeButtonField . . . . .	2795, 2810, 2834, 2881, 2892
\_int_eval:n . . . . .	776	\MakeChoiceField . . . . .	2770
\_int_max:nn . . . . .	777	\MakeFieldObject . . . . .	2435
\_int_new:N . . . . .	469	\MakeRadioField . . . . .	2752
iow commands:		\MakeTextField . . . . .	2631
\_iow_newline: . . . . .	2974, 2980, 2993	\mbox . . . . .	1204
<b>K</b>			
\kern . . . . .	3496	menu (hypersetup key) . . . . .	10, 1470
keys commands:		menuborder (hypersetup key) . . . . .	13
\_keys_define:nn . . . . .	171, 179, 197, 235, 384, 952, 968, 975, 987, 996, 1021, 1036, 1055, 1082, 1101, 1224, 1238, 1250, 1273, 1279, 1294, 1323, 1354, 1367, 1378, 1392, 1402, 1411, 1424, 1451, 1472, 1477, 2033, 2115, 2131, 2164, 2202, 2216, 2234, 2262, 2318, 2333, 2346, 2366	menuborderstyle (hypersetup key) . . . . .	14, 1080
\_l_keys_key_str . . . . .	176, 2188, 2431	menucolor (hypersetup key) . . . . .	966
\_keys_set:nn . . . . .	168, 205, 224, 251, 281, 305, 323, 343, 394, 405, 1384, 1391, 2239, 2266, 2269, 2270, 2271, 2272, 2327, 2356, 2431	mode commands:	
\_keys_set_known:nn . . . . .	2434	\_mode_if_horizontal:TF . . . . .	594, 616
\kvssetkeys . . . . .	167, 2522	\_mode_leave_vertical: . . . . .	248, 278, 302, 320, 340, 643, 726, 785, 832, 869, 2764, 2792, 2809, 2831, 2873, 2916
<b>L</b>			
\label . . . . .	8, 13	msg commands:	
\LayoutCheckField . . . . .	2914	\_msg_error:nn . . . . .	415, 2806, 2891
		\_msg_info:nnn . . . . .	2141, 2146
		\_msg_line_context: . . . . .	68
		\_g_msg_module_name_prop . . . . .	11
		\_msg_new:nnn . . . . .	36, 43, 48, 52, 56, 63, 70, 77, 84, 91, 97, 104, 113, 123, 134
		\_msg_new:nnnn . . . . .	12, 22
		\_msg_warning:nn . . . . .	155, 1398
		\_msg_warning:nnn . . . . .	164, 647, 887
		\_msg_warning:nnnn . . . . .	1242, 1283, 1520, 1564, 1598, 1621, 1683, 1712, 1746, 1761,

1774, 1787, 1812, 1845, 1861, 1888, 1919, 1952, 1966, 1993, 2027, 2109	
\msg_warning:nnnnn . . . . .	175,
1314, 1348, 1459, 1540, 1577, 1664, 1697, 1722, 1798, 1825, 1874, 1979, 2006, 2256, 2373, 2392, 2411, 2422	
<b>N</b>	
naturalnames (hypersetup key) . . . . .	<a href="#">1424</a>
nesting (hypersetup key) . . . . .	<a href="#">13</a>
\newcommand . . . . .	<a href="#">149</a> , <a href="#">3468</a> , <a href="#">3470</a> , <a href="#">3472</a>
\newcount . . . . .	<a href="#">2576</a> , <a href="#">2760</a>
\NewDocumentCommand . . . . .	<a href="#">2435</a>
\NewExpandableDocumentCommand . . . . .	<a href="#">371</a>
\newlength . . . . .	<a href="#">3487</a>
\noexpand . . . . .	<a href="#">2658</a> , <a href="#">2659</a> , <a href="#">2660</a> , <a href="#">2661</a> , <a href="#">3039</a>
\nolinkurl . . . . .	<a href="#">3</a> , <a href="#">4</a>
<b>O</b>	
ocgcolorfile (hypersetup key) . . . . .	<a href="#">1220</a>
ocgcolorlink (hypersetup key) . . . . .	<a href="#">1220</a>
ocgcolorlinks (hypersetup key) . . . . .	<a href="#">1220</a>
ocgcolormenu (hypersetup key) . . . . .	<a href="#">1220</a>
ocgcolorrun (hypersetup key) . . . . .	<a href="#">1220</a>
ocgcolorurl (hypersetup key) . . . . .	<a href="#">1220</a>
<b>P</b>	
pageanchor (hypersetup key) . . . . .	<a href="#">1424</a>
\paperwidth . . . . .	<a href="#">3</a>
\PassOptionsToPackage . . . . .	<a href="#">436</a> , <a href="#">1416</a> , <a href="#">1421</a>
pdf commands:	
\pdf_bdcobject:nn . . . . .	<a href="#">1206</a> , <a href="#">1209</a>
\pdf_destination:nn . . . . .	<a href="#">26</a> , <a href="#">141</a> , <a href="#">611</a>
\pdf_destination:nnnn . . . . .	<a href="#">604</a>
\pdf_emc: . . . . .	<a href="#">1208</a> , <a href="#">1214</a>
\pdf_link_user:nnn . . . . .	<a href="#">2767</a>
\pdf_name_from_unicode_e:n . . . . .	<a href="#">230</a> , <a href="#">756</a> , <a href="#">871</a>
\pdf_object_if_exist:nTF . . . . .	<a href="#">754</a>
\pdf_object_new:nn . . . . .	<a href="#">1128</a> ,
<a href="#">1129</a> , <a href="#">1130</a> , <a href="#">1131</a> , <a href="#">2961</a> , <a href="#">2962</a> , <a href="#">2963</a>	
\pdf_object_ref:n . . . . .	<a href="#">142</a> , <a href="#">766</a> ,
<a href="#">1134</a> , <a href="#">1136</a> , <a href="#">1158</a> , <a href="#">1159</a> , <a href="#">1162</a> , <a href="#">1166</a> ,	
<a href="#">1171</a> , <a href="#">1176</a> , <a href="#">1181</a> , <a href="#">2553</a> , <a href="#">2555</a> , <a href="#">3019</a>	
\pdf_object_ref_last: . . . . .	<a href="#">787</a> , <a href="#">2361</a>
\pdf_object_unnamed_write:nn . . . . .	<a href="#">786</a> , <a href="#">2357</a>
\pdf_object_write:nn . . . . .	<a href="#">1132</a> ,
<a href="#">1138</a> , <a href="#">1148</a> , <a href="#">1160</a> , <a href="#">2964</a> , <a href="#">3006</a> , <a href="#">3013</a>	
\pdf_pageobject_ref:n . . . . .	<a href="#">1905</a> , <a href="#">1930</a> , <a href="#">2591</a>
\pdf_string_from_unicode:nnN . . . . .	<a href="#">555</a>
\pdf_version: . . . . .	<a href="#">1568</a> , <a href="#">1687</a> , <a href="#">1716</a> , <a href="#">1750</a> , <a href="#">1765</a> ,
<a href="#">1791</a> , <a href="#">1816</a> , <a href="#">1849</a> , <a href="#">1865</a> , <a href="#">1970</a> , <a href="#">1997</a>	
\pdf_version_compare:NnTF . . . . .	<a href="#">1558</a> , <a href="#">1649</a> , <a href="#">1671</a> , <a href="#">1707</a> , <a href="#">1740</a> , <a href="#">1755</a> ,
<a href="#">1781</a> , <a href="#">1806</a> , <a href="#">1832</a> , <a href="#">1855</a> , <a href="#">1960</a> , <a href="#">1987</a>	
\pdf_version_compare_p:Nn . . . . .	<a href="#">823</a> , <a href="#">1221</a> , <a href="#">1270</a>
\pdf_version_major: . . . . .	<a href="#">152</a> , <a href="#">1222</a> , <a href="#">1245</a> , <a href="#">1271</a> , <a href="#">1287</a>
\pdf_version_minor: . . . . .	<a href="#">151</a> , <a href="#">1245</a> , <a href="#">1287</a>
pdfannot commands:	
\pdfannot_box:nnnn . . . . .	<a href="#">2626</a> , <a href="#">2737</a> , <a href="#">2796</a> , <a href="#">2847</a> , <a href="#">2882</a> , <a href="#">2946</a>
\pdfannot_box_ref_last: . . . . .	<a href="#">2482</a> , <a href="#">2498</a>
\pdfannot_dict_put:nnn . . . . .	<a href="#">727</a> , <a href="#">787</a> , <a href="#">833</a> , <a href="#">872</a> , <a href="#">1012</a> ,
<a href="#">1047</a> , <a href="#">1071</a> , <a href="#">1093</a> , <a href="#">1117</a> , <a href="#">1299</a> , <a href="#">1330</a>	
\pdfannot_dict_remove:nn . . . . .	<a href="#">1002</a> ,
<a href="#">1042</a> , <a href="#">1063</a> , <a href="#">1088</a> , <a href="#">1109</a> , <a href="#">1307</a> , <a href="#">1340</a>	
\pdfannot_link:nnn . . . . .	<a href="#">728</a> , <a href="#">788</a> , <a href="#">834</a> , <a href="#">873</a>
\pdfannot_link_goto_begin:nw . . . . .	<a href="#">660</a>
\pdfannot_link_goto_end: . . . . .	<a href="#">665</a>
\pdfannot_link_margin:n . . . . .	<a href="#">8</a> , <a href="#">1602</a>
\c_pdfannot_link_types_seq . . . . .	<a href="#">523</a>
pdfauthor (hypersetup key) . . . . .	<a href="#">2129</a>
\pdfbookmark . . . . .	<a href="#">3473</a>
pdfborder (hypersetup key) . . . . .	<a href="#">13</a>
pdfborderstyle (hypersetup key) . . . . .	<a href="#">14</a> , <a href="#">1080</a>
pdfcreationdate (hypersetup key) . . . . .	<a href="#">14</a> , <a href="#">2200</a>
pdfcreator (hypersetup key) . . . . .	<a href="#">2129</a>
\pdfdest . . . . .	<a href="#">11</a>
pdfdict commands:	
\pdfdict_new:n . . . . .	<a href="#">542</a> , <a href="#">705</a> , <a href="#">742</a> , <a href="#">801</a> , <a href="#">858</a>
\pdfdict_put:nnn . . . . .	<a href="#">222</a> , <a href="#">229</a> , <a href="#">543</a> , <a href="#">706</a> , <a href="#">707</a> , <a href="#">718</a> ,
<a href="#">721</a> , <a href="#">743</a> , <a href="#">744</a> , <a href="#">756</a> , <a href="#">763</a> , <a href="#">773</a> , <a href="#">783</a> ,	
<a href="#">802</a> , <a href="#">803</a> , <a href="#">814</a> , <a href="#">829</a> , <a href="#">859</a> , <a href="#">860</a> , <a href="#">870</a> ,	
<a href="#">1628</a> , <a href="#">1629</a> , <a href="#">1633</a> , <a href="#">1634</a> , <a href="#">2370</a> , <a href="#">2380</a> ,	
<a href="#">2384</a> , <a href="#">2387</a> , <a href="#">2389</a> , <a href="#">2408</a> , <a href="#">2417</a> , <a href="#">2419</a>	
\pdfdict_remove:nn . . . . .	<a href="#">219</a> , <a href="#">826</a> , <a href="#">1638</a> , <a href="#">1639</a>
\pdfdict_use:n . . . . .	<a href="#">727</a> , <a href="#">786</a> , <a href="#">833</a> , <a href="#">839</a> , <a href="#">872</a> , <a href="#">877</a> , <a href="#">2359</a>
pdfencoding (hypersetup key) . . . . .	<a href="#">1392</a>
\pdfescapestring . . . . .	<a href="#">3028</a>
pdffile commands:	
\pdffile_embed_file:nnn . . . . .	<a href="#">758</a>
PDFForm internal commands:	
\PDFForm_hypName . . . . .	<a href="#">3067</a> , <a href="#">3070</a> , <a href="#">3074</a> , <a href="#">3077</a>
pdfinfo (hypersetup key) . . . . .	<a href="#">2262</a>
pdfkeywords (hypersetup key) . . . . .	<a href="#">2129</a>
pdflang (hypersetup key) . . . . .	<a href="#">14</a> , <a href="#">2115</a>
pdflinkmargin (hypersetup key) . . . . .	<a href="#">14</a>
pdfmanagement commands:	
\pdfmanagement_add:nn . . . . .	<a href="#">1611</a>

<code>\pdfmanagement_add:nnn</code> . . . . .	
. . . . . 578, 1158, 1159, 1181, 1487,	
1512, 1528, 1532, 1552, 1560, 1589,	
1644, 1654, 1678, 1692, 1704, 1709,	
1734, 1742, 1757, 1783, 1808, 1840,	
1857, 1903, 1928, 1942, 1962, 1989,	
2018, 2124, 2142, 2147, 2158, 2176,	
2187, 2211, 2225, 2243, 2342, 2361,	
2528, 2545, 2552, 2554, 2556, 2562	
<code>\pdfmanagement_if_active_p:</code> . . . 414	
<code>\pdfmanagement_remove:nn</code> . . . . .	
. . . 1484, 1508, 1516, 1536, 1548,	
1573, 1585, 1593, 1607, 1616, 1660,	
1675, 1694, 1731, 1770, 1795, 1821,	
1836, 1870, 1900, 1925, 1938, 1947,	
1975, 2002, 2014, 2023, 2121, 2151,	
2170, 2208, 2222, 2339, 2352, 2566	
<code>\pdfmanagement_show:n</code> . . . . . 2529	
pdfmeta commands:	
<code>\pdfmeta_standard_item:n</code> . . . . .	
. . . . . 2324, 2329, 2330	
<code>\pdfmeta_standard_verify:nnTF</code> . . .	
. . . . . 867, 2783, 2874	
<code>\pdfmeta_standard_verify:nTF</code> . . .	
. . . . . 2558, 3135	
pdfmoddate (hypersetup key) . . . . . <a href="#">14</a> , <a href="#">2200</a>	
pdfproducer (hypersetup key) . . . . . <a href="#">2129</a>	
pdfremotestartview (hypersetup key) . . . <a href="#">11</a>	
pdfstartview (hypersetup key) . . . . . <a href="#">11</a>	
<code>\pdfstringdef</code> . . . . . <a href="#">8</a> , <a href="#">9</a> , <a href="#">22</a> , <a href="#">28</a> , <a href="#">3084</a>	
<code>\pdfstringdefDisableCommands</code> . . . . .	
. . . . . 372, 373, 374	
pdfsubject (hypersetup key) . . . . . <a href="#">2129</a>	
pdftitle (hypersetup key) . . . . . <a href="#">2129</a>	
pdftrapped (hypersetup key) . . . . . <a href="#">2234</a>	
pdfversion (hypersetup key) . . . . . <a href="#">1392</a>	
pdfview (hypersetup key) . . . . . <a href="#">11</a> , <a href="#">2033</a>	
pdfxform commands:	
<code>\pdfxform_if_exist:nTF</code> . . . . .	
. . . . . 2723, 2835, 2919	
<code>\pdfxform_new:nnn</code> . . . . .	
. . . 2437, 2725, 2838, 2842, 2921, 2933	
<code>\pdfxform_ref:n</code> . . . . .	
. . . 2747, 2748, 2854, 2855, 3166, 3167	
<code>\phantom</code> . . . . . 2942	
<code>\phantomsection</code> . . . . . <a href="#">12</a> , <a href="#">18</a> , <a href="#">3556</a>	
plainpages (hypersetup key) . . . . . <a href="#">1424</a>	
prg commands:	
<code>\prg_do_nothing:</code> . . . . . 399, 410	
<code>\prg_generate_conditional_</code>	
variant:Nnn . . . . . 447	
<code>\prg_new_eq_conditional:NNn</code> . . . 446	
prop commands:	
<code>\prop_const_from_keyval:Nn</code> . . . . .	
. . . . . 480, 488, 3617, 3632,	
3647, 3662, 3677, 3692, 3708, 3723	
<code>\prop_gput:Nnn</code> . . . . . 11, 2455, 2462	
<code>\prop_if_empty:NTF</code> . . . . . 2531	
<code>\prop_item:Nn</code> . . . . . 2548	
<code>\prop_map_inline:Nn</code> . . . . .	
. . . . . 930, 994, 1034, 1061,	
1069, 1080, 1107, 1115, 1184, 1292,	
1328, 1338, 1382, 1470, 2526, 2533	
<code>\prop_new:N</code> . . . . . 2441, 2442	
<code>\protect</code> 252, 282, 306, 324, 344, 452, 681,	
732, 739, 793, 799, 843, 849, 892, 3489	
<code>\providecommand</code> . . . . .	
. . . . . 147, 148, 2469, 2472, 2636, 3024	
<code>\ProvidesFile</code> . . . . . 3	
	<b>R</b>
<code>\ReadBookmarks</code> . . . . . 3478	
ref commands:	
<code>\ref_if_exist:nn</code> . . . . . 446	
<code>\ref_label:nn</code> . . . . . 442	
<code>\ref_value:nn</code> . . . . . <a href="#">25</a> , <a href="#">463</a>	
<code>\refstepcounter</code> . . . . . <a href="#">12</a>	
regex commands:	
<code>\regex_const:Nn</code> . . . . . 529	
<code>\regex_extract_once:NnNTF</code> 1883, 1914	
<code>\relax</code> . . . . . 681, 732, 739,	
793, 799, 843, 849, 892, 2639, 2640,	
2681, 3021, 3068, 3072, 3142, 3146,	
3249, 3250, 3258, 3262, 3289, 3293,	
3373, 3381, 3408, 3416, 3540, 3549	
<code>\RemoveFromHook</code> . . . . . 1492	
<code>\RenewDocumentCommand</code> . . . . . 158	
<code>\RequirePackage</code> . . . . . 6, 18, 431, 3476	
run (hypersetup key) . . . . . <a href="#">10</a> , <a href="#">1470</a>	
runborder (hypersetup key) . . . . . <a href="#">13</a>	
runborderstyle (hypersetup key) . . . <a href="#">14</a> , <a href="#">1080</a>	
runcolor (hypersetup key) . . . . . <a href="#">966</a>	
	<b>S</b>
<code>\selectfont</code> . . . . . 2732, 2929, 2941	
seq commands:	
<code>\seq_const_from_clist:Nn</code> . . . . . 478	
<code>\seq_count:N</code> . . . . . 2042, 2068	
<code>\seq_get_right:NN</code> . . . . . 2044	
<code>\seq_item:Nn</code> . . . . . 1885,	
1916, 2038, 2086, 2088, 2098, 2103	
<code>\seq_map_inline:Nn</code> . . . . .	
. . . 515, 519, 523, 966, 1267, 1365, 2543	
<code>\seq_new:N</code> . . . . . 468	
<code>\seq_put_right:Nn</code> . . . . .	
. . . . . 2075, 2076, 2077, 2535	



<code>\seq_set_split:Nnn</code> .....	2037	<code>\@currDisplay</code> .....	2651, 2719
<code>\seq_sort:Nn</code> .....	2537	<code>\@currValue</code> .....	2717, 2747
<code>\setbox</code> .....	365	<code>\@currentHref</code> .....	
<code>\setlength</code> .....	3506, 3577	.....	3510, 3523, 3534, 3564, 3581
<code>\setpdflinkmargin</code> .....	8	<code>\@curropt</code> ..	2636, 2646, 2649, 2713, 2716
<code>\settowidth</code> .....	2651	<code>\@empty</code> .....	2605, 2621, 3089,
<code>\show</code> .....	2711	3093, 3097, 3101, 3105, 3109, 3113,	
sort commands:		3117, 3121, 3125, 3269, 3300, 3301	
<code>\sort_return_same:</code> .....	2541	<code>\@endForm</code> .....	2571
<code>\sort_return_swapped:</code> .....	2540	<code>\@esphack</code> .....	443
<code>\space</code> .....	456, 2719, 3251, 3268,	<code>\@firstofone</code> ..	2475, 3030, 3035, 3052
3269, 3299, 3300, 3374, 3406, 3409		<code>\@firstoftwo</code> .....	3571
<code>\spacefactor</code> .....	594, 616	<code>\@for</code> .....	2647, 2714
<code>\stockwidth</code> .....	3	<code>\@gobbletwo</code> .....	3520
str commands:		<code>\@hyper@launch</code> .....	37, 853
<code>\c_colon_str</code> .....	853	<code>\@ifnextchar</code> .....	260
<code>\c_hash_str</code> .....	723	<code>\@ifundefined</code> .....	3044
<code>\c_percent_str</code> .....	724	<code>\@latex@warning</code> .....	453
<code>\str_case:nnTF</code> .....	2137, 3431	<code>\@mainaux</code> .....	2468, 2471, 2491, 2510
<code>\str_case_e:nnTF</code> .....	2038	<code>\@mkboth</code> .....	3520
<code>\str_compare:nNnTF</code> .....	2539	<code>\@nil</code> .....	3042, 3055, 3061
<code>\str_gset:Nn</code> .....	991	<code>\@part</code> .....	3547, 3554
<code>\str_gset_eq:NN</code> .....	565	<code>\@pdfauthor</code> .....	22
<code>\str_head:n</code> .....	2245, 2250	<code>\@pdfborder</code> .....	147
<code>\str_if_eq:nnTF</code> .....		<code>\@pdfborderstyle</code> .....	148
... 601, 2156, 2174, 2184, 2324, 3133		<code>\@savsf</code> .....	594, 616
<code>\str_if_eq_p:nn</code> .....	821, 1222, 1271	<code>\@schapter</code> .....	3517, 3518
<code>\str_lowercase:n</code> .....		<code>\@secondoftwo</code> .....	3573
..... 2038, 2179, 2227, 2246, 2251		<code>\@sect</code> .....	3568, 3569
<code>\str_new:N</code> .....	470, 475, 476, 514	<code>\@spart</code> .....	3560, 3561
<code>\str_set:Nn</code> .....	547	<code>\@ssect</code> .....	3503, 3504
<code>\str_set_convert:Nnnn</code> .....	8	<code>\@tempdima</code> ..	2645, 2652, 2654, 2691,
<code>\str_set_eq:NN</code> .....	567	2692, 2693, 2697, 2698, 2699, 2700	
<code>\str_tail:n</code> .....	2246, 2251	<code>\@tempdimb</code> .....	2651, 2652
<code>\str_uppercase:n</code> .....	2239, 2245, 2250	<code>\@typeset@protect</code> .....	3489
<code>\string</code> .....	2469,	<code>\@unexpandable@protect</code> .....	
2472, 2493, 2511, 3044, 3048, 3059		.....	252, 282, 306, 324, 344
<code>\subpdfbookmark</code> .....	3469	<code>\BKM@color</code> .....	424
sys commands:		<code>\c@secnumdepth</code> .....	3537, 3555, 3570
<code>\c_sys_backend_str</code> .....	3431	<code>\calc@bm@number</code> .....	3481
		<code>\check@bm@number</code> .....	3480
		<code>\define@key</code> .....	418
		<code>\Fld@additionalactions</code> ..	3131, 3201,
		3235, 3271, 3308, 3355, 3395, 3424	
		<code>\Fld@align</code> .....	3160, 3246, 3321
		<code>\Fld@altname</code> ..	3068, 3070, 3142, 3144
		<code>\Fld@annotflags</code> .....	3155,
		3208, 3241, 3277, 3316, 3365, 3400	
		<code>\Fld@annotnames</code> .....	3140, 3157,
		3210, 3243, 3279, 3318, 3367, 3402	
		<code>\Fld@bcolor</code> .....	3179, 3181, 3262,
		3264, 3293, 3295, 3329, 3342, 3344	
		<code>\fld@bcolor</code> .....	3250

## T

T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> commands:

<code>\@BOOKMARK</code> .....	3475
<code>\@CheckBox</code> .....	2897
<code>\@ChoiceMenu</code> .....	2637
<code>\@Form</code> .....	2520
<code>\@PushButton</code> .....	2775
<code>\@Reset</code> .....	2864
<code>\@Submit</code> .....	2816
<code>\@TextField</code> .....	2602
<code>\@bookmarksopenlevel</code> .....	1503
<code>\@bsphack</code> .....	441
<code>\@chapter</code> .....	3529, 3530

\Fld@bordercolor	3175, 3177, 3219, 3228, 3230, 3249, 3258, 3260, 3289, 3291, 3327, 3338, 3340, 3373, 3381, 3383, 3408, 3416, 3418	2693, 2738, 2752, 2770, 2788, 2818, 2829, 2866, 2878, 2902, 2912, 2947
\Fld@borderstyle	3161, 3214, 3247, 3283, 3322, 3371, 3422	\Fld@X@additionalactions
\Fld@borderwidth	2698, 2699, 3161, 3214, 3247, 3283, 3322, 3371, 3422	3088, 3133, 3136
\Fld@calculate@code	2484, 3101, 3103	\Form@action
\Fld@calculate@sortkey	2495	\G@refundefinedtrue
\Fld@cbsymbol	3183	\H@old@part
\Fld@charsize	2697, 3187, 3268, 3299, 3350, 3406	\H@old@chapter
\Fld@checkequals	2649, 2716	\H@old@sect
\Fld@choices	3270	\H@old@spart
\Fld@color	3188, 3190, 3269, 3300, 3351	\H@old@ssect
\Fld@default	2604, 2621, 2639, 2682, 2711, 2712, 2900, 3301, 3305, 3306, 3353	\href@
\Fld@flags	3159, 3212, 3245, 3281, 3320, 3369, 3404	\href@split
\Fld@format@code	3093, 3095	\Hy@abspage
\Fld@height	2607, 2628, 2631, 2642, 2700, 2739, 2752, 2770, 2819, 2867, 2903, 2948	\Hy@activeanchorfalse
\Fld@keystroke@code	3089, 3091	\Hy@activeanchortrue
\Fld@listcount	2710, 2718, 2753, 2760	\Hy@AtBeginDocument
\Fld@mappingname	3072, 3074, 3146, 3148	\Hy@bookmarksnumberedfalse
\Fld@maxlen	3356, 3358	\Hy@bookmarksnumberedtrue
\Fld@menulength	2644, 2650, 2697	\Hy@bookmarksopenfalse
\Fld@name	2603, 2638, 2776, 2899, 3067, 3141	\Hy@bookmarksopentru
\Fld@onblur@code	3109, 3111	\Hy@bookmarkstype
\Fld@onclick@code	3234	\Hy@chapapp
\Fld@onenter@code	3121, 3123	\Hy@colorlink
\Fld@onexit@code	3125, 3127	\Hy@currentbookmarklevel
\Fld@onfocus@code	3105, 3107	\Hy@DisableOption
\Fld@onmousedown@code	3113, 3115	\Hy@drafttrue
\Fld@onmouseup@code	3117, 3119	\Hy@escapeform
\Fld@pageobjref	2573, 2587, 3024, 3156, 3209, 3242, 3278, 3317, 3366, 3401	2625, 2722, 2766, 2794, 2833, 2880, 2918, 3026, 3033, 3036
\Fld@radiosymbol	3297	\Hy@escapestring
\Fld@rotation	3171, 3173, 3217, 3224, 3226, 3248, 3254, 3256, 3285, 3287, 3325, 3334, 3336, 3372, 3377, 3379, 3407, 3412, 3414	3028, 3030, 3035, 3038, 3039, 3042, 3043, 3052, 3055, 3091, 3095, 3099, 3103, 3107, 3111, 3115, 3119, 3123, 3127, 3183, 3234, 3297, 3353, 3354, 3391
\Fld@submitflags	3393	\Hy@finaltrue
\Fld@validate@code	3097, 3099	\Hy@FormObjects
\Fld@value	2605, 2621, 2640, 2681, 2682, 3354	\Hy@gttemp
\Fld@width	2606, 2620, 2627, 2631, 2641, 2680, 2691,	\Hy@href
		\Hy@href@nextactionraw
		\Hy@href@page
		\Hy@hypertextnamesfalse
		\Hy@hypertextnamestrue
		\Hy@linkfileprefix
		\Hy@linktoc
		\Hy@localanchornamefalse
		\Hy@localanchornamestrue
		\Hy@MakeCurrentHref
		\Hy@MakeCurrentHrefAuto
		3505, 3521, 3532, 3562, 3576
		\Hy@naturalnamesfalse
		\Hy@naturalnamestrue
		\Hy@next
		\Hy@numberline
		150, 3463



\Hy@org@chapter	3529, 3544	\HyField@PDFChoices	2763
\Hy@OutlineName	3479	\HyField@SetKeys	2615, 2656, 2659, 2678, 2778, 2821, 2869, 2904
\Hy@OutlineRerunCheck	3477	\HyPat@ObjRef	3435, 3442, 3449, 3456
\Hy@pageanchorfalse	1443	\hyper@link	268
\Hy@pageanchortrue	1442	\hyper@anchor	618
\Hy@pdfmajorversion	152	\hyper@anchorend	618, 3498, 3523, 3534, 3564
\Hy@pdfminorversion	151	\hyper@anchorstart	618, 3498, 3523, 3534, 3564
\Hy@pdfstringtrue	29, 560	\hyper@link	33, 668
\Hy@pdfversion	3079	\hyper@linkend	33, 696
\Hy@plainpagesfalse	1447	\hyper@linkfile	312, 747
\Hy@plainpagestrue	1446	\hyper@linklaunch	36, 330, 805, 856
\Hy@pstringdef	570, 3059	\hyper@linknamed	37, 160, 862
\Hy@PutCatalog	576	\hyper@linkstart	33, 684
\Hy@raisedlink	3497, 3522, 3533, 3563	\hyper@linkurl	294, 358, 709
\Hy@RestoreLastskip	615	\hyper@normalise	260, 290, 307, 325, 352, 365
\Hy@safe@activestrue	725, 2487	\HyPL@Labels	578, 588
\Hy@SaveLastskip	595	\HyPL@StorePageLabel	576
\Hy@secnum@part	3550, 3552, 3555	\HyPsd@SanitizeForOutFile	3466
\Hy@SectionAnchorHref	3488, 3510, 3581	\if@filesw	2467, 2490, 2509
\Hy@SectionHShift	3487, 3496, 3506, 3577	\ifFld@combo	2665, 2695
\Hy@StepCount	2650, 2718	\ifFld@hidden	2620, 2680
\Hy@temp	2488, 2503, 3041, 3042	\ifFld@multiline	2608
\Hy@unicodefalse	3081	\ifFld@popdown	2666
\Hy@VerboseAnchor	596	\ifFld@radio	2662, 2685
\Hy@verbosefalse	1406	\ifHy@implicit	3483
\Hy@VerboseLinkStart	672, 688	\ifHy@pdfescapeform	3027, 3037
\Hy@VerboseLinkStop	679, 702, 735, 795, 845, 883	\ifHy@unicode	3080
\Hy@verbosetrue	1405	\kv@set@family@handler	163
\Hy@VersionChecked	143	\ltx@empty	2444, 2445, 2484, 2571, 2572, 2573, 2604
\Hy@WrapperDef	592	\ltx@gobble	2447
\Hy@xspace@end	676, 734, 794, 844, 882	\ltx@gobbletwo	2448
\HyAnn@AbsPageLabel	2572, 2578, 2624, 2721, 2765, 2793, 2832, 2879, 2917	\ltx@ifUndefined	3528, 3538
\HyAnn@Count	2576, 2577, 2580, 2581, 2582, 2583, 2584, 2585, 2589, 2593	\ltx@one	2580
\HyField@AddToFields	2478, 2632, 2754, 2771, 2802, 2858, 2887, 2951	\ltx@ReturnAfterFi	3060
\HyField@afields	2444	\ltx@zero	2577
\HyField@AfterAuxOpen	2449, 2475, 2508	\m@ne	3537, 3552
\HyField@AuxAddToCoFields	2448, 2460, 2472, 2493	\OBJ@OCG@view	149
\HyField@AuxAddToFields	2447, 2453, 2469, 2511	\p@	2654
\HyField@cofields	2445	\pdf@ifdraftmode	2523
\HyField@FlagsCheckBox	2909	\PDF@SetupDoc	144
\HyField@FlagsChoice	2690	\PDFForm@Check	2918, 2950, 3152
\HyField@FlagsPushButton	2785, 2825, 2876	\PDFForm@List	2766, 2769, 3239
\HyField@FlagsRadioButton	2686	\PDFForm@Name	2618, 2679, 2782, 2908, 3066
\HyField@FlagsSubmit	2826	\PDFForm@Push	2794, 2800, 3205
\HyField@FlagsText	2619	\PDFForm@Radio	2722, 2742, 3274
		\PDFForm@Reset	2880, 2886, 3398
		\PDFForm@Submit	2833, 2852, 3363
		\PDFForm@Text	2625, 2630, 3313

\protected@edef	644	\tl_if_eq:NnTF	2045
\strip@pt	3187, 3268, 3299, 3350, 3406	\tl_if_exist:NTF	3228, 3338, 3342
\toks@	3508, 3514, 3579, 3585	\tl_if_exist_p:N	3219, 3327, 3329
\tw@	3509, 3514, 3580, 3585	\tl_if_head_eq_charcode:nNTF	898, 915
\url@	365, 368	\tl_new:N	190, 191, 192, 193, 467, 471, 472, 473, 474, 497, 498, 499, 500, 501, 502, 503, 504, 505, 513, 618
\Url@def	366	\tl_set:Nn	210, 211, 223, 255, 258, 285, 288, 347, 350, 377, 507, 508, 509, 510, 511, 512, 624, 633, 674, 690, 752, 1881, 1885, 1889, 1912, 2047, 2050, 2057, 2060, 2061, 2062, 2063, 2064, 2065, 2070, 2074, 2110
\Url@HyperHook	366	\tl_set_eq:NN	651
\XR@ext	1426	\tl_to_str:N	754, 761, 766
\z@	365, 2645, 3248, 3254, 3285, 3372, 3377, 3407, 3412, 3550	\tl_to_str:n	18, 19, 20, 31, 455
\zref@labelbylist	2582	token commands:	
\zref@labelbyprops	2581	\token_to_str:N	27
\zref@refused	2583	\toks	3509, 3514, 3580, 3585
\texorpdfstring	29		
text commands:			
\text_expand:n	752		
\text_purify:n	8, 547		
\textsf	2840, 2844		
\the	588, 2581, 2582, 2583, 2584, 2585, 2589, 2593, 2693, 2697, 2700, 3514, 3585		
\theH...	12		
\thepage	456		
\ThisShouldNotHappen	3046		
\title	2		
tl commands:			
\c_space_tl	150, 1135, 1166, 1171, 1176, 2747, 2748, 2755, 2970, 2972, 2982, 2984, 2986, 2989, 2991, 3187, 3190, 3322, 3350, 3351		
\tl_const:Nn	477		
\tl_gput_right:Nn	588		
\tl_gset:Nn	1896, 1916, 1920		
\tl_if_blank:nTF	771, 2135, 2168, 2206, 2220, 2337, 2350		
\tl_if_empty:NTF	645, 1482, 3175, 3179, 3188, 3351		
\tl_if_empty:nTF	217, 1000, 1040, 1059, 1086, 1105, 1673, 1729, 1834, 2119		
\tl_if_empty_p:N	1898, 1923		
		<b>U</b>	
		unicode (hypersetup key)	1392
		\Url	3, 5, 194
		url (hypersetup key)	10, 1470
		\url	3–6, 22, 338, 365
		urlborder (hypersetup key)	13
		urlborderstyle (hypersetup key)	14, 1080
		urlcolor (hypersetup key)	966
		\urldef	5, 22, 365, 367
		\Urlfont	6
		\UrlSpecial	4
		use commands:	
		\use:n	851
		<b>V</b>	
		verbose (hypersetup key)	1402
		<b>W</b>	
		\write	2468, 2471, 2491, 2510
		<b>X</b>	
		\x	2657, 2677, 3513, 3515, 3584, 3586