

# The `hyperref-generic` module

## A generic driver for `hyperref`

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

Version 0.95k, released 2022-03-16

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 `pdflatex`, `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
\DocumentMetadata % activates it
{
  %% options
  %% e.g. pdf version, backend:
  % pdfversion=1.7,
  % backend = dvipdfmx
}
```

Starting with L<sup>A</sup>T<sub>E</sub>X-2022-06-01 (or a suitable `latex-dev`) loading the testphase package is no longer required and this can be simplified to

```
\DocumentMetadata
{
  %% 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.

---

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

## 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.
- 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 an 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 `\DocumentMetadata` should be used:

```
\cs{DocumentMetadata}{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/graphicx`, `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 `graphicx` 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 `cmyk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmyk` (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_colorruncolor_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorruncolor_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 reenable 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	<code>urlbordercolor = [rgb]{1,1,0}</code>
color expression	<code>urlbordercolor = red!50!blue</code>
command	<code>urlbordercolor = \mycolor</code>

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 `luatex` 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`
- `hypertextnames`
- `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 `\DocumentMetadata`

**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 `\DocumentMetadata`:

```
\DocumentMetadata{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 `cmYk` (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> )	Key	Values	comment / example
<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 dvips it is possible to change links locally, pdf <sub>l</sub> atex and lua <sub>l</sub> atex work by page, with dvipdfmx 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 set the language in <code>\DocumentMetadata</code> instead.		

## File I

# hyperref-generic driver implementation

```

1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2022-03-16 v0.95k %
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 \cs_if_exist:NTF \DocumentMetadata
13 {

```

```

14 \msg_new:nnnn
15 { hyp }
16 { missing-resource-management }
17 { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
18 {
19     Activate~it~with \\
20     \tl_to_str:n{\DocumentMetadata{<options>}}\\
21     before~\tl_to_str:n{\documentclass}
22 }
23 }
24 {
25     \msg_new:nnnn
26     { hyp }
27     { missing-resource-management }
28     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
29     {
30         Activate~it~with \\
31         \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
32         \tl_to_str:n{\DocumentMetadata{<options>}}\\
33         before~\tl_to_str:n{\documentclass}
34     }
35 }

```

The pdfversion should be set in \DocumentMetadata

```

36 \msg_new:nnnn
37 { hyp }
38 { pdfversion-disabled }
39 {
40     This~hyperref~driver~ignores~the~pdfversion~key!\\
41     Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata
42 }
43 {
44     For example:\\
45     \tl_to_str:n
46     {
47         \DocumentMetadata { pdfversion=1.7 }
48     }
49 }

```

A generic message for ignored keys.

```

50 \msg_new:nnn
51 { hyp }
52 { key-dropped }
53 {
54     This~hyperref~driver~ignores~the~key~#1!\\
55     Please~check~the~documentation.
56 }

```

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

```

57 \msg_new:nnn
58 { hyp }
59 { pdfa-no-push-button }
60 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
61
62 \msg_new:nnn

```

```

63 { hyp }
64 { pdfa-no-reset-button }
65 { PDF/A:~Reset~action~is~prohibited }

```

pdf/A message for not allowed Named actions

```

66 \msg_new:nnn
67 { hyp }
68 { pdfa-no-named-action }
69 { PDF/A:~Named~action~#1~is~prohibited }

```

A message if the destination name is empty.

```

70 \msg_new:nnn
71 { hyp }
72 { empty-destination-name }
73 {
74   Empty~destination~name,\\
75   using~'~#1'
76 }

```

A message if the destination check fails

```

77 \msg_new:nnn
78 { hyp }
79 { invalid-destination-value }
80 {
81   Invalid~value~'~#1'~of~'~#2'   \\
82   is~replaced~by~'Fit'~\msg_line_context:.
83 }

```

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

```

84 \msg_new:nnn
85 { hyp }
86 { ignore-deprecated-or-unknown-option-in-pdf-version }
87 {
88   Option~'~#1'~is~unknown~or~deprecated~in\\
89   pdf~version~#2.~Ignored.
90 }
91 \msg_new:nnn
92 { hyp }
93 { ignore-deprecated-or-unknown-value-in-pdf-version }
94 {
95   Value~'~#1'~is~unknown~or~deprecated~in\\
96   pdf~version~#2.~Ignored.
97 }
98 \msg_new:nnn
99 { hyp }
100 { replace-deprecated-or-unknown-value-in-pdf-version }
101 {
102   Value~'~#1'~is~unknown~or~deprecated~in\\
103   pdf~version~#2. Value~'~#3'~is used instead.
104 }

```

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:

```

105 \msg_new:nnn
106 { hyp }
107 { unknown-key }

```



```

108 {
109     unknown~key~#2~of~module~'#1'~set~to~'#3'.
110 }
111 \msg_new:nnn
112 { hyp }
113 { unknown-key-to-Hyp }
114 {
115     ignored~in~family~Hyp~unknown~key~#1.
116 }

```

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

```

117 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
118 \msg_new:nnn
119 { hyp }
120 { unknown-choice }
121 {
122     Value~'#3'~is~invalid~for~key~'#1'.\\
123     The~key~accepts~only~the~choices\\
124     \clist_map_function:nN { #2 } \__hyp_clist_display:n
125 }
126
127 \msg_new:nnn
128 { hyp }
129 { unknown-choice+empty }
130 {
131     Value~'#3'~is~invalid~for~key~'#1'.\\
132     The~key~accepts~only~the~choices\\
133     \clist_map_function:nN { #2 } \__hyp_clist_display:n
134     An~empty~value~removes~the~setting.
135 }
136
137 \msg_new:nnn
138 { hyp }
139 { no-bool }
140 {
141     Value~'#2'~is~invalid~for~key~'#1'.\\
142     The~key~accepts~only~the~choices\\
143     *~true\\
144     *~false \\
145     *~and~an~empty~value~which~removes~the~setting.\\
146     No~value~is~equivalent~to~using~'true'.
147 }

```

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

```

148 \msg_new:nnn
149 { hyp }
150 { empty-info-value }
151 {
152     Empty~value~for~key~#1.\\
153     This~isn't~honored~by~all~backends.
154 }

```

## 2 Variants

```

155 \cs_generate_variant:Nn\pdf_destination:nn {nf}
156 \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

```

157 \chardef\Hy@VersionChecked=1 %don't check the version!
158 %\cs_set_protected:Npn \PDF@SetupDoc{}
159 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

---

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

`\hypercalcbp`

```

160 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

(End definition for `\hypercalcbp`. This function is documented on page 18.)

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

```

161 \providecommand\@pdfborder{}
162 \providecommand\@pdfborderstyle{}
163 \newcommand\OBJ@OCG@view {} % needed in hyperref
164 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark

```

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

```

165 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
166 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
167 \legacy_if:nT { Hy@setpdfversion }
168 {
169   \msg_warning:nn { hyp }{ pdfversion-disabled }
170 }
171 \Hy@DisableOption{pdfversion}

```

`\Acrobatmenu` should use the new internal link command

```

172 \RenewDocumentCommand \Acrobatmenu { m m }
173 {
174   \hyper@linknamed {#1} {#2}
175 }

```

`\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!

```

176 % TODO should go at some time ...
177 % \kv@set@family@handler{Hyp}
178 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
179 \cs_set_protected:Npn \hypersetup #1
180 {
181   \kvsetkeys{Hyp} {#1}
182   \keys_set:nn { hyp / setup }{ #1 }
183 }
184 % TODO for now unknown keys should only give warnings.

```

```

185 \keys_define:nn { hyp / setup }
186 {
187   unknown .code:n =
188   {
189     \msg_warning:nnxxx { hyp } { unknown-key }
190     { hyp~/~setup }{ \l_keys_key_str } { #1 }
191   }
192 }

```

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.

```

193 \keys_define:nn { hyp / setup }
194 {
195   next-anchor .code:n =
196   {
197     \AddToHookNext{__hyp/dest/make}
198     {\Hy@MakeCurrentHref{#1}}
199   }
200 }

```

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`)

```

201
202 \bool_new:N \l__hyp_href_url_encode_bool
203 \bool_new:N \l__hyp_href_url_ismap_bool
204 \tl_new:N \l__hyp_href_url_protocol_tl
205 \tl_new:N \l__hyp_href_pdf_destination_tl
206 \tl_new:N \l__hyp_href_pdf_page_tl
207 \tl_new:N \l__hyp_href_run_parameter_tl
208 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
209
210
211 \keys_define:nn { hyp / href }
212 {
213   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
214   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
215   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
216   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
217   ,pdfremotestartview .code:n =
218   {
219     \keys_set:nn { hyp / setup }
220     { pdfremotestartview = #1 }
221   }
222   ,page .code:n =
223   {
224     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
225     \tl_set:Nn \Hy@href@page {#1}
226   }
227   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
228   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
229   ,nextactionraw .code:n =

```

```

230 { %perhaps some safety match later, see hyperref code
231   \tl_if_empty:nTF {#1}
232   {
233     \pdfdict_remove:nn{l_hyp/annot/A}{Next}
234   }
235   {
236     \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
237     \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
238     \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
239   }
240 }
241 ,afrelationship .code:n =
242 {
243   \pdfdict_put:nnx
244   { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
245 }
246
247 }
248
249 \keys_define:nn { hyp / setup }
250 {
251   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
252   ,href / urlencode .default:n = {true}
253   ,href / urlencode .initial:n = {false}
254   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
255   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
256   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
257 }
258
259 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
260
261 \DeclareRobustCommand*{\href}[1][ ]{%
262   \mode_leave_vertical:
263   \hook_use:n{cmd/href/before}
264   \group_begin:
265   \keys_set:nn { hyp / href } {#1}
266   \let\protect\@unexpandable@protect
267   \bool_if:NTF \l__hyp_href_url_encode_bool
268   {
269     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
270   }
271   {
272     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
273   }
274   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
275 }
276
277 \begingroup
278   \catcode'\$=6 %
279   \catcode'\#=12 %
280   \gdef\href@${1}\expandafter\href@split$1#\$\}%
281   \gdef\href@split$1#$2#$3\\$4{%
282     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
283   \endgroup

```

```

284     \hook_use:n{cmd/href/after}
285   }%
286 \endgroup
287
288 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
289
290 \DeclareRobustCommand*{\hrefurl}[1] []
291 {
292   \mode_leave_vertical:
293   \hook_use:n{cmd/href/before}
294   \group_begin:
295   \keys_set:nn { hyp / href } {#1}
296   \let\protect\@unexpandable@protect
297   \bool_if:NTF \l__hyp_href_url_encode_bool
298   {
299     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
300   }
301   {
302     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
303   }
304   \hyper@normalise\__hyp_href_url_aux:nn}
305
306 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
307 {
308   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
309   \group_end:
310   \hook_use:n{cmd/href/after}
311 }
312
313 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
314 \DeclareRobustCommand*{\hrefpdf}[1] []
315 {
316   \mode_leave_vertical:
317   \hook_use:n{cmd/hrefpdf/before}
318   \group_begin:
319   \keys_set:nn { hyp / href } {#1}
320   \let\protect\@unexpandable@protect
321   \hyper@normalise\__hyp_href_pdf_aux:nn
322 }
323
324 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
325 {
326   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
327   \group_end:
328   \hook_use:n{cmd/hrefpdf/after}
329 }
330
331 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
332 \DeclareRobustCommand*{\hrefrun}[1] []
333 {
334   \mode_leave_vertical:
335   \hook_use:n{cmd/hrefrun/before}
336   \group_begin:
337   \keys_set:nn { hyp / href } {#1}

```

```

338 \let\protect\@unexpandable@protect
339 \hyper@normalise\__hyp_href_run_aux:nn
340 }
341
342 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
343 {
344   \exp_args:Nnno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
345   \group_end:
346   \hook_use:n{cmd/hrefrun/after}
347 }
348
349
350 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
351
352 \DeclareRobustCommand*\url{[1] []
353 {
354   \mode_leave_vertical:
355   \hook_use:n{cmd/url/before}
356   \group_begin:
357   \keys_set:nn {hyp / href } {#1}
358   \let\protect \@unexpandable@protect
359   \bool_if:NTF \l__hyp_href_url_encode_bool
360   {
361     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
362   }
363   {
364     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
365   }
366   \hyper@normalise\__hyp_href_url_aux:n
367 }
368
369 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
370 {
371   \exp_args:Nno
372   \hyper@linkurl{\__hyp_href_url_format: {#1}}
373   {\l__hyp_href_url_protocol_tl#1}
374   \group_end:
375   \hook_use:n{cmd/url/after}
376 }
377

```

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

```

378
379 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
380 \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
381 % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
382 \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
383 #2}
384

```

make the new commands compatible with `\pdfstringdef`:

```

385 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
386 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
387 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}

```

```
388 \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.

```
389 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
390 {
391   %\tl_set:cn {@#1}{#2}
392   \AddToDocumentProperties[hyperref]{#1}{#2}
393 }
394 \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.

```
395 \hook_new:n{hyp/link/cite}
396 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
397 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
398 \keys_define:nn { hyp / setup }
399 {
400   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init
401   ,citebordercolor
402   .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_init
403 }
404 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
405 {
406   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
407   {
408     \keys_set:nn { hyp / setup }
409     {
410       linkcolor      = hyp/color/cite
411     }
412   }
413 \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
414 }
415 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
416 {
417   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
418   {
419     \keys_set:nn { hyp / setup }
420     {
421       linkbordercolor = hyp/color/citeborder
422     }
423   }
424 \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
425 }
426
```

## 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

```

427 \bool_lazy_and:nnF
428 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
429 { \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

```

430 \AddToHook { package/bookmark/after}
431 {
432   \define@key{BKM}{color}
433   {
434     \__hyp_color_set:ne {__hyp/tmpa}{#1}
435     \color_export:nVN
436       {__hyp/tmpa}
437       \g__hyp_bordercolormodel_str
438       \BKM@color
439   }
440 }
441 \legacy_if:nT { Hy@bookmarks }
442 {
443   \AddToHook{begindocument/before}[hyperref/bookmark]
444   {
445     \RequirePackage{bookmark}
446   }
447 }
448 \legacy_if:nT { Hy@draft}
449 {
450   \PassOptionsToPackage{draft}{bookmark}
451 }

```

## 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 452 %
\__hyp_ref_check:nn 453 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn 454 {
455   \@bsphack
456   \ref_label:nn{#1}{#2}
457   \@esphack
458 }
459 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```



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

```
460 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
461 \prg_generate_conditional_variant:Nnn \__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.

```
462 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
463 {
464   \__hyp_ref_if_exist:nnF {#1}{#2}
465   {
466     \protect\G@refundefinedtrue
467     \@latex@warning
468     {
469       Reference~‘\tl_to_str:n {#1}’~with~attribute~‘\tl_to_str:n {#2}’~
470       on~page~\thepage~\space undefined
471     }
472   }
473 }
```

```
474 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}
```

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

```
475 \cs_new:Npn \__hyp_ref_value:nn #1 #2
476 {
477   \ref_value:nn{#1}{#2}
478 }
479 \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
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str
480 \box_new:N \l__hyp_tmpa_box
481 \tl_new:N \l__hyp_tmpa_tl
482 \seq_new:N \l__hyp_tmpa_seq
483 \int_new:N \l__hyp_tmpa_int
484 \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.

TODO: document and check use!

```
\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
\__hyp_text_tmpa_str_uu\g__hyp_text_tmpa_str
485 \tl_new:N \l__hyp_dest_name_tmpa_tl
486 \tl_new:N \l__hyp_uri_tmpa_tl
487 \tl_new:N \l__hyp_filename_tmpa_tl
488 \tl_new:N \l__hyp_para_tmpa_tl
489 \str_new:N \l__hyp_text_tmpa_str
490 \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.

```
491 \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  
`\c__hyp_map_annot_hyp_prop` annot names to hyperref names and back.  
`\c__hyp_map_hyp_annot_prop`

```
492 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
493 {url,link,file,menu,run}
494 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
495 {
496   URI    = url,
497   GoTo   = link,
498   GoToR  = file,
499   Named  = menu,
500   Launch = run
501 }
502 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
503 {
504   url    = URI,
505   link   = GoTo,
506   file   = GoToR,
507   menu   = Named,
508   run    = Launch
509 }
510
```

(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  
`\g__hyp_dest_pdfstartview_tl` the current and remote files. The instruction is in “PDF format” but without the leading  
`\l__hyp_dest_pdfremotestartview_tl` slash!

```
511 \tl_new:N \g__hyp_dest_pdfstartpage_tl
512 \tl_new:N \g__hyp_dest_pdfstartview_tl
513 \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
514 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
515 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
516 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
517 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
518 \tl_new:N \l__hyp_text_enc_file_print_tl
519 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

520
521 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
522 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
523 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
524 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
525 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
526 \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.

```

527 \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*)

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`

```

528 \str_new:N \g__hyp_bordercolormodel_str

```

(End definition for `\g__hyp_bordercolormodel_str`.)

## 7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrun_bool`

`\l_hyp_annot_colormenu_bool`

```

529 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

530 {

```

```

531   \bool_new:c {l_hyp_annot_color#1_bool}

```

```

532 }

```

(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_ocgcolorrun_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.

```

533 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

534 {

```

```

535   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}

```

```

536 }

```

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

`\l_hyp_annot_Named_bool`

This booleans are used to disable some link types while keeping others.

```

537 \seq_map_inline:Nn \c_pdfannot_link_types_seq

```

```

538 {

```

```

539   \bool_new:c {l__hyp_annot_#1_bool}

```

```

540   \bool_set_true:c {l__hyp_annot_#1_bool}

```

```

541 }

```

(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.

```
542 \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.

```
543 \regex_const:Nn \c__hyp_dest_startview_regex
544 {
545   \A\ *
546   (?:
547     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
548     |
549     (?:Fit\b|FitB\b)
550     |
551     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
552     |
553     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
554   )
555 }
```

*(End definition for \c\_\_hyp\_dest\_startview\_regex.)*

## 7.7 PDF dictionaries

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

```
556 \pdfdict_new:n {l__hyp_page/Trans}
557 \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`

```
558 \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`

```

559 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
560 {
561     \str_set:Nx #2 {\text_purify:n { #1 } }
562 }

```

*(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`

```

563 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
564 {
565
566 }

```

*(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

utf8/string	(lit) (utf8/string)
utf8/string-raw	lit (utf8/string)
utf8/URI	(percent encoded url)
utf8/URI-raw	percent encoded url
utf16/hex	<HEX> (utf16/hex)
utf16/hex-raw	HEX (utf16/hex)
utf16/string	(lit) (utf16/string)
utf16/string-raw	lit (utf16/string)

`\__hyp_text_string_from_unicode:nN`

```

567 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
568 {
569     \pdf_string_from_unicode:nVN { #1 } #2 #2
570 }

```

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

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

`\__hyp_text_pdfstring:nnN`

```

571 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
572 {
573     \group_begin:
574     \Hy@pdfstringtrue
575     \hook_use:n {hyp/text/pdfstring}
576     \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
577     \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
578     \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
579     \str_gset_eq:NN \g__hyp_text_tmpa_str \l__hyp_text_tmpa_str
580     \group_end:
581     \str_set_eq:NN #3 \g__hyp_text_tmpa_str
582 }
583 \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
584 \cs_new_protected:Npn \Hy@pstringdef #1 #2
585 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }
This is a special version for info keys:
```

```
\_hyp\_text\_pdfstring\_info:nN
```

```
586 \cs_new_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
587 {
588   \_hyp\_text\_pdfstring:nonN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
589 }
```

```
(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
```

```
590 \cs_new_protected:Npn \_hyp\_PageLabels\_gpush:
591 {
592   \pdfmanagement\_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
```

```

593 }
594
595 \def\Hy@PutCatalog #1 {}
596
597
598 \legacy_if:nT { Hy@pdfpagelabels }
599 {
600   \cs_set_protected:Npn \HyPL@StorePageLabel #1
601   {
602     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
603     \__hyp_PageLabels_gppush:
604   }
605 }

```

(End definition for `\__hyp_PageLabels_gppush:`, `\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

606 \Hy@WrapperDef \__hyp_destination:nn #1 #2
607 {
608   \mode_if_horizontal:T { \@savsf\spacefactor }
609   \Hy@SaveLastskip      %defined in hyperref
610   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
611   \__hyp_text_pdfstring:xoN
612   { \HyperDestNameFilter{#1} }

```

```

613     { \l__hyp_text_enc_dest_tl }
614     \l__hyp_tmpa_tl
615 \str_if_eq:nnTF {#2} {fitrbox}
616 {
617     \exp_args:NV
618     \pdf_destination:nnnn \l__hyp_tmpa_tl
619     { \box_wd:N \l__hyp_dest_box }
620     { \box_ht:N \l__hyp_dest_box }
621     { \box_dp:N \l__hyp_dest_box }
622 }
623 {
624     \exp_args:NV
625     \pdf_destination:nf
626     { \l__hyp_tmpa_tl }
627     { #2 }
628 }
629 \Hy@RestoreLastskip %defined in hyperref
630 \mode_if_horizontal:T { \spacefactor\@savsf }
631 }

```

(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
\hyper@anchorend
hyp/anchor
\l_hyp_current_dest_name_tl
632 \tl_new:N\l_hyp_current_dest_name_tl
633 \hook_new:n{hyp/anchor}
634 \cs_new_protected:Npn \hyper@anchor #1
635 {
636     \exp_args:NnV
637     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
638     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
639     \hook_use:n{hyp/anchor}
640 }
641
642 \cs_new_protected:Npn \hyper@anchorstart #1
643 {
644     \Hy@activeanchortrue
645     \exp_args:NnV
646     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
647     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
648     \hook_use:n{hyp/anchor}
649 }
650
651 \cs_new_protected:Npn \hyper@anchorend
652 {
653     \Hy@activeanchorfalse
654 }

```

(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:

```

655 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
656 {
657   \mode_leave_vertical:
658   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
659   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
660   {
661     \msg_warning:nnx
662     { hyp }
663     { empty-destination-name }
664     { \c__hyp_dest_undefined_tl }
665     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
666   }
667   {
668     \__hyp_text_pdfstring:xoN
669     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
670     { \l__hyp_text_enc_dest_tl }
671     \l__hyp_dest_name_tmpa_tl
672   }
673   \exp_args:No
674   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
675 }
676
677 \cs_new_protected:Npn \__hyp_link_goto_end:
678 {
679   \pdfannot_link_goto_end:
680 }
```

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.

---

<code>\hyper@linkstart</code>	<code>\hyper@linkstart{&lt;context&gt;}{&lt;destination name&gt;}</code>
<code>\hyper@linkend</code>	<code>\hyper@linkend</code>

---

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\_hyp\_current\_dest\_name\_tl` so that the hook code can make use of it

```

681
682 \cs_new_protected:Npn \hyper@link #1 #2 #3 {%#1 context, #2=destination name, #3 content
683 {
684   \bool_if:NTF \l__hyp_annot_GoTo_bool
685   {
686     \Hy@VerboseLinkStart{#1}{#2}
687     \group_begin:
688     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
689     \hook_use:n {hyp/link/#1}
690     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
691     \__hyp_link_goto_end:
692     \group_end:
693     \Hy@VerboseLinkStop
694   }
695   {\let\protect\relax#3}}
696 }
697
698 \cs_new_protected:Npn \hyper@linkstart #1 #2 {%#1 context, #2=destination name
699 {
700   \bool_if:NT \l__hyp_annot_GoTo_bool
701   {
702     \Hy@VerboseLinkStart{#1}{#2}% only for debug
703     \group_begin:
704     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
705     \hook_use:n {hyp/link/#1}
706     \__hyp_link_goto_begin:nw {#2}
707   }
708 }
709
710 \cs_new_protected:Npn \hyper@linkend
711 {
712   \bool_if:NT \l__hyp_annot_GoTo_bool
713   {
714     \__hyp_link_goto_end:
715     \group_end:
716     \Hy@VerboseLinkStop
717   }
718 }
```

### 10.3 URI links

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

```

719 \pdfdict_new:n {l_hyp/annot/A/URI}
```

```

720 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
721 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
722
723 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
724 {
725   \bool_if:NTF \l__hyp_annot_URI_bool
726   {
727     \group_begin:
728     \__hyp_text_pdfstring:xoN
729     { #2}
730     { \l__hyp_text_enc_uri_print_tl }
731     \l__hyp_uri_tmpa_tl
732     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
733     \bool_if:NT \l__hyp_href_url_ismap_bool
734     {
735       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
736     }
737     \cs_set_eq:NN \# \c_hash_str
738     \cs_set_eq:NN \% \c_percent_str
739     \Hy@safe@activetrue
740     \mode_leave_vertical:
741     \pdfannot_dict_put:nxx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
742     \pdfannot_link:nxx { URI }
743     {
744     }
745     {
746       \let\protect\relax
747       #1
748       \Hy@xspace@end
749       \Hy@VerboseLinkStop %where is the start??
750     }
751     \group_end:
752   }
753   {{\let\protect\relax#1}}
754 }
755

```

## 10.4 GoToR Links files

```

756 \pdfdict_new:n {l_hyp/annot/A/GoToR}
757 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
758 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
759
760
761 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
762 {
763   \bool_if:NTF \l__hyp_annot_GoToR_bool
764   {
765     \group_begin:
766     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
767     \exp_args:Nx
768     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
769     {
770       \pdfdict_put:nxx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap

```

```

771         \exp_args:Nnox
772         \pdffile_embed_file:nnn
773         {}
774         {\l__hyp_filename_tmpa_tl }
775         {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
776     }
777     \pdfdict_put:nnx
778     {l_hyp/annot/A/GoToR}
779     {F}
780     {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
781     \__hyp_text_pdfstring:nnN
782     { #3 }
783     { \l__hyp_text_enc_dest_print_tl }
784     \l__hyp_dest_name_tmpa_tl
785     \tl_if_blank:eTF {#3}
786     {
787         \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
788         {
789             [
790                 \int_eval:n
791                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
792                 /\l__hyp_dest_pdfremotestartview_tl
793             ]
794         }
795     }
796     {
797         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
798     }
799     \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>

```

800     \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
801     \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
802     \pdfannot_link:nnx %expansion??
803     { GoToR }
804     {
805     }
806     {
807         \let\protect\relax
808         #1\Hy@xspace@end
809         \Hy@VerboseLinkStop %where is the start??
810     }
811     \group_end:
812 }
813 {\let\protect\relax#1}}
814 }

```

## 10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

815 \pdfdict_new:n {l_hyp/annot/A/Launch}
816 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
817 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
818

```

```

819 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
820 {
821   \bool_if:NTF \l__hyp_annot_Launch_bool
822   {
823     \group_begin:
824     \__hyp_text_pdfstring:nnN
825       { #1 }
826       { \l__hyp_text_enc_file_print_tl }
827       \l__hyp_filename_tmpa_tl
828     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
829     \__hyp_text_pdfstring:noN
830       { #3 }
831       { \l__hyp_text_enc_para_print_tl }
832       \l__hyp_para_tmpa_tl
833     \bool_if:nTF
834       {
835         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
836         ||
837         \pdf_version_compare_p:Nn > {1.9}
838       }
839       {
840         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
841       }
842       {
843         \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
844         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
845       }
846     \mode_leave_vertical:
847     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
848     \pdfannot_link:nxn
849       { Launch }
850       {
851         % /A
852         % <<
853         % \pdfdict_use:n {l_hyp/annot/A/Launch}
854         % >>
855       }
856       {
857         \let\protect\relax
858         #2\Hy@xspace@end
859         \Hy@VerboseLinkStop %where is the start??
860       }
861     \group_end:
862   }
863   {\let\protect\relax#2}}
864 }

```

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

```

865 \use:x
866 { % filename, anchor text, linkname
867   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
868 }
869 {

```

```

870 \hyper@linklaunch {#1}{#2}{#3}
871 }

```

## 10.6 Named links (menu)

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

```

872 \pdfdict_new:n {l_hyp/annot/A/Named}
873 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
874 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
875
876 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
877 {
878   \bool_if:NTF \l__hyp_annot_Named_bool
879   {
880     \group_begin:
881     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
882     {
883       \mode_leave_vertical:
884       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
885       {\pdf_name_from_unicode_e:n{#1}}
886       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
887       \pdfannot_link:nnx { Named }
888       {
889         % /A
890         % <<
891         % \pdfdict_use:n { l_hyp/annot/A/Named }
892         % >>
893       }
894       {
895         #2
896         \Hy@xspace@end
897         \Hy@VerboseLinkStop
898       }
899     }
900     {
901       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
902       #2
903     }
904     \group_end:
905   }
906   {{\let\protect\relax#2}}
907 }
908

```

## 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.

909 \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.

```

910 \cs_new_protected:Npn \_hyp_color_select:n #1
911 {
912   \tl_if_head_eq_charcode:nNTF {#1}[ %]
913   {
914     \_hyp_color_select_aux:wn #1
915   }
916   {
917     \color_select:n {#1}
918   }
919 }
920
921 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
922 {
923   \color_select:nn {#1}{#2}
924 }
925
926 \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.

```

927 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
928 {
929   \tl_if_head_eq_charcode:nNTF {#2}[ %]
930   {
931     \_hyp_color_set_aux:nwn { #1 } #2
932   }
933   {
934     \color_set:nn {#1} {#2}
935   }
936 }
937
938 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
939 {
940   \color_set:nnn {#1}{#2}{#3}
941 }
942
943 \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.

```

944 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
945 {
946   \hook_gput_code:nnn
947     {pdfannot/link/#2/begin}
948     {hyp/color}
949     {
950       \bool_if:cT { l_hyp_annot_color#1_bool }
951       {
952         \group_begin:
953         \color_select:n { hyp/color/#1}
954       }
955     }
956   \hook_gput_code:nnn
957     {pdfannot/link/#2/end}
958     {hyp/color}
959     {
960       \bool_if:cT { l_hyp_annot_color#1_bool }
961       {
962         \group_end:
963       }
964     }
965 }
```

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

```

966 \keys_define:nn { hyp / setup }
967 {
968   ,colorlinks .meta:n =
969   {
970     ,pdfborder={0~0~0}
971     ,pdfborderstyle=
972     ,colorurl =#1
973     ,colorlink =#1
974     ,colorrurl =#1
975     ,colormenu =#1
976     ,colorfile =#1
977   }
978   ,colorlinks .default:n = {true}
979 }
```

`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*)

```

980 \seq_map_inline:Nn \c__hyp_annot_types_seq
981 {
982   \keys_define:nn { hyp / setup }
983   {
984     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
985     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
986   }
987 }
988 }
```



```

989 \keys_define:nn { hyp / setup }
990 {
991   ,allcolors .meta:n =
992   {
993     ,urlcolor=#1
994     ,linkcolor=#1
995     ,runcolor=#1
996     ,filecolor=#1
997     ,menucolor=#1
998   }
999   ,allcolors .value_required:n = true
1000 }

```

## 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*)

```

1001 \keys_define:nn { hyp / setup }
1002 {
1003   ,bordercolormodel .choices:nn =
1004   {rgb,cmyk}
1005   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1006   ,bordercolormodel .initial:n = {rgb}
1007 }

1008 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1009 {
1010   \keys_define:nn { hyp / setup }
1011   {
1012     #1bordercolor .code:n =
1013     {
1014       \tl_if_empty:nTF { ##1 }
1015       {
1016         \pdfannot_dict_remove:nn
1017         {link/#2}
1018         { C }
1019       }
1020       {
1021         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1022         \color_export:nVN
1023         {hyp/color/#1border}
1024         \g__hyp_bordercolormodel_str
1025         \l__hyp_tmpa_tl
1026         \pdfannot_dict_put:nnx
1027         {link/#2}
1028         { C }
1029         { [\l__hyp_tmpa_tl] }
1030       }
1031     }
1032   }
1033 }

```

```

1034
1035 \keys_define:nn { hyp / setup }
1036 {
1037   ,allbordercolors .meta:n =
1038   {
1039     ,linkbordercolor=#1
1040     ,urlbordercolor =#1
1041     ,filebordercolor=#1
1042     ,menubordercolor=#1
1043     ,runbordercolor =#1
1044   }
1045   ,allbordercolors .value_required:n = true
1046 }
1047

```

### 11.3.2 Borderwidth and -arc

```

1048 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1049 {
1050   \keys_define:nn { hyp / setup }
1051   {
1052     #1border .code:n =
1053     {
1054       \tl_if_empty:nTF { ##1 }
1055       {
1056         \pdfannot_dict_remove:nn
1057         {link/#2}
1058         { Border }
1059       }
1060       {
1061         \pdfannot_dict_put:nnn
1062         {link/#2}
1063         { Border }
1064         { [##1] }
1065       }
1066     }
1067   }
1068 }
1069 \keys_define:nn { hyp / setup }
1070 {
1071   ,pdfborder .code:n =
1072   {
1073     \tl_if_empty:nTF { #1 }
1074     {
1075       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1076       {
1077         \pdfannot_dict_remove:nn
1078         {link/##2}
1079         { Border }
1080       }
1081     }
1082     {
1083       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1084       {

```

```

1085         \pdfannot_dict_put:nnn
1086         {link/##2}
1087         { Border }
1088         { [#1] }
1089     }
1090 }
1091 }
1092 ,pdfborder .initial:n = {0~0~1},
1093 }

```

### 11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1094 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1095 {
runborderstyle (setup key) 1096   \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1097   {
menuborderstyle (setup key) 1098     #1borderstyle .code:n =
1099     {
1100       \tl_if_empty:nTF { ##1 }
1101       {
1102         \pdfannot_dict_remove:nn
1103         {link/#2}
1104         { BS }
1105       }
1106       {
1107         \pdfannot_dict_put:nnn
1108         {link/#2}
1109         { BS }
1110         { <<##1>> }
1111       }
1112     }
1113   }
1114 }
1115 \keys_define:nn { hyp / setup }
1116 {
1117   ,pdfborderstyle .code:n =
1118   {
1119     \tl_if_empty:nTF { #1 }
1120     {
1121       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1122       {
1123         \pdfannot_dict_remove:nn
1124         {link/##2}
1125         { BS }
1126       }
1127     }
1128     {
1129       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1130       {
1131         \pdfannot_dict_put:nnn
1132         {link/##2}
1133         { BS }

```

```

1134         { <<#1>> }
1135     }
1136 }
1137 }
1138 ,pdfborderstyle .initial:n = {},
1139 }

```

## 11.4 ocgcolorlinks

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.

```

1140 \cs_new_protected:Npn \__hyp_ocg_init:
1141 {
1142     \pdf_object_new:nn { __hyp/OCG/View } { dict }
1143     \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1144     \pdf_object_new:nn { __hyp/OCG/config } { dict }
1145     \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1146     \pdf_object_write:nx { __hyp/OCG/refarray }
1147     {
1148         \pdf_object_ref:n { __hyp/OCG/View }
1149         \c_space_tl
1150         \pdf_object_ref:n { __hyp/OCG/Print }
1151     }
1152     \pdf_object_write:nn { __hyp/OCG/View }
1153     {
1154         /Type/OCG
1155         /Name(View)
1156         /Usage
1157         <<
1158         /Print <</PrintState/OFF>>~
1159         /View <</ViewState/ON >>~
1160         >>
1161     }
1162     \pdf_object_write:nn { __hyp/OCG/Print }
1163     {
1164         /Type/OCG
1165         /Name(Print)
1166         /Usage
1167         <<
1168         /Print <</PrintState/ON>>~
1169         /View <</ViewState/OFF>>~
1170         >>
1171     }
1172     \pdfmanagement_add:nxx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View} }
1173     \pdfmanagement_add:nxx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1174     \pdf_object_write:nx { __hyp/OCG/config }
1175     {
1176         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1177         /AS[
1178         <<
1179         /Event/View

```

```

1180         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1181         /Category[/View]
1182     >>
1183     <<
1184         /Event/Print
1185         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1186         /Category[/Print]
1187     >>
1188     <<
1189         /Event/Export
1190         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1191         /Category[/Print]
1192     >>
1193     ]
1194 }
1195 \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1196 \cs_gset:Npn \__hyp_ocg_init: {}
1197 }

```

(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.

```

1198 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1199 {
1200     \hook_gput_code:nnn
1201     {pdfannot/link/#2/begin}
1202     {hyp/ocg}
1203     {
1204         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1205         {
1206             \__hyp_ocg_init:
1207             \group_begin:
1208             \hbox_set:Nw \l__hyp_tmpa_box
1209         }
1210     }
1211     \hook_gput_code:nnn
1212     {pdfannot/link/#2/end}
1213     {hyp/ocg}
1214     {
1215         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1216         {
1217             \hbox_set_end:
1218             \mbox
1219             {
1220                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1221                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1222                 \pdf_emc:
1223                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1224                 \group_begin:
1225                 \color_select:n { hyp/color/#1 }
1226                 \box_use_drop:N \l__hyp_tmpa_box
1227                 \group_end:
1228                 \pdf_emc:
1229             }

```

```

1230         \group_end:
1231     }
1232 }
1233 }

```

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

```

1234 \bool_lazy_or:nnTF
1235 { \pdf_version_compare_p:Nn > {1.4} }
1236 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1237 {
1238     \keys_define:nn { hyp / setup }
1239     {
1240         ,_ocgcolorlinks .meta:n =
1241         {
1242             ocgcolorlink=#1,
1243             ocgcolorurl=#1,
1244             ocgcolorfile=#1,
1245             ocgcolorrun=#1,
1246             ocgcolormenu=#1
1247         }
1248         ,_ocgcolorlinks .default:n = true
1249     }
1250 }
1251 {
1252     \keys_define:nn { hyp / setup }
1253     {
1254         ,_ocgcolorlinks .code:n =
1255         {
1256             \msg_warning:nnxx
1257             { hyp }
1258             { ignore-deprecated-or-unknown-option-in-pdf-version }
1259             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1260         }
1261     }
1262 }
1263
1264 \keys_define:nn { hyp / setup }
1265 {
1266     ,ocgcolorlinks .choice:
1267     ,ocgcolorlinks / true .meta:n =
1268     {
1269         pdfborder      = {0~0~0},
1270         pdfborderstyle = {},
1271         colorlinks     = false,
1272         _ocgcolorlinks = true
1273     }
1274     ,ocgcolorlinks / false .meta:n =
1275     {
1276         _ocgcolorlinks = false
1277     }
1278     ,ocgcolorlinks .default:n = {true}
1279 }
1280
1281 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

1282 {
1283   \bool_lazy_or:nnTF
1284   { \pdf_version_compare_p:Nn > {1.4} }
1285   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1286   {
1287     \keys_define:nn { hyp / setup }
1288     {
1289       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1290     }
1291   }
1292   {
1293     \keys_define:nn { hyp / setup }
1294     {
1295       ,ocgcolor#1 .code:n=
1296       {
1297         \msg_warning:nnxx
1298         { hyp }
1299         { ignore-deprecated-or-unknown-option-in-pdf-version }
1300         { ocgcolor#1 }
1301         { \pdf_version_major:.\pdf_version_minor: }
1302       }
1303     }
1304   }
1305 }

```

## 11.5 Highlighting

This keys set what happens if you click on a link

```

1306 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1307 {
1308   \keys_define:nn { hyp / setup }
1309   {
1310     ,#1highlight .choices:nn =
1311     { /I, /N, /O, /P }
1312     {
1313       \pdfannot_dict_put:nnn
1314       {link/#2}
1315       { H }
1316       { ##1 }
1317     }
1318     ,#1highlight / .code:n =
1319     {
1320       \pdfannot_dict_remove:nn
1321       {link/#2}
1322       { H }
1323     }
1324     ,#1highlight / unknown .code:n =
1325     {
1326       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1327       { pdfhighlight }
1328       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1329       { \exp_not:n {##1} }
1330     }
1331   }

```

```

1332     }
1333   }
1334 }
1335
1336
1337 \keys_define:nn { hyp / setup }
1338 {
1339   ,pdfhighlight .choices:nn =
1340   { /I, /N, /O, /P }
1341   {
1342     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1343     {
1344       \pdfannot_dict_put:nnn
1345       {link/####2}
1346       { H }
1347       { #1 }
1348     }
1349   }
1350   ,pdfhighlight / .code:n =
1351   {
1352     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1353     {
1354       \pdfannot_dict_remove:nn
1355       {link/##2}
1356       { H }
1357     }
1358   }
1359   ,pdfhighlight .initial:n = {/I},
1360   ,pdfhighlight / unknown .code:n =
1361   {
1362     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1363     { pdfhighlight }
1364     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1365     { \exp_not:n {#1} }
1366   }
1367 }

```

## 11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1368 \keys_define:nn { hyp / setup }
hideurl (setup key) 1369 {
hidefile (setup key) 1370   hidelinks .meta:n =
hiderun (setup key) 1371   {
hidemenu (setup key) 1372     ,colorlinks      = false
1373     ,ocgcolorlinks = false
1374     ,pdfborder      = { 0~0~0 }
1375     ,pdfborderstyle=
1376   }
1377 }
1378
1379 \seq_map_inline:Nn \c__hyp_annot_types_seq

```



```

1380 {
1381   \keys_define:nn { hyp / setup }
1382   {
1383     hide#1 .meta:n =
1384     {
1385       ,color#1      = false
1386       ,ocgcolor#1   = false
1387       ,#1border     = { 0~0~0 }
1388       ,#1borderstyle =
1389     }
1390   }
1391 }

```

## 11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1392 \keys_define:nn { hyp / setup }
1393 {
1394   colorscheme .code:n =
1395   {
1396     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1397     {
1398       \keys_set:nn { hyp / setup }
1399       {
1400         ##1 = ##2
1401       }
1402     }
1403   }
1404 }
1405 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

## 12 Keys

### 12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode (setup key)`

`pdfencoding (setup key)`

`pdfversion (setup key)`

```

1406 \keys_define:nn { hyp / setup }
1407 {
1408   ,unicode      .code:n = {}
1409   ,pdfencoding .code:n = {}
1410   ,pdfversion .code:n =
1411   {
1412     \msg_warning:nn { hyp }{ pdfversion-disabled }
1413   }
1414 }
1415 %

```

## 12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
  debug (setup key) 1416 \keys_define:nn { hyp / setup }
  draft (setup key) 1417 {
  final (setup key) 1418   ,verbose .choice:
1419   ,verbose / true .code:n = { \Hy@verbosetrue}
1420   ,verbose / false .code:n = { \Hy@verbosefalse}
1421   ,verbose .default:n = {true}
1422   ,debug .meta:n = {verbose=#1}
1423   ,debug .default:n = {true}
1424 }
1425 \keys_define:nn { hyp / setup }
1426 {
1427   ,draft .code:n =
1428   {
1429     \Hy@drafttrue
1430     \PassOptionsToPackage{draft}{bookmark}
1431   }
1432   ,final .code:n =
1433   {
1434     \Hy@finaltrue
1435     \PassOptionsToPackage{final}{bookmark}
1436   }
1437 }

extension (setup key)
hypertextnames (setup key) 1438 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1439 {
  pageanchor (setup key) 1440   ,extension .tl_set:N = \XR@ext
  linktoc (setup key) 1441   ,extension .initial:n= pdf
  linktocpage (setup key) 1442   ,hypertextnames .choice:
  plainpages (setup key) 1443   ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1444   ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1445   ,hypertextnames .default:n = {true}
1446   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1447   ,localanchorname .choice:
1448   ,localanchorname / true .code:n = { \Hy@localanchorname true }
1449   ,localanchorname / false .code:n = { \Hy@localanchorname false }
1450   ,localanchorname .default:n = {true}
1451   ,naturalnames .choice:
1452   ,naturalnames / true .code:n = { \Hy@naturalnames true}
1453   ,naturalnames / false .code:n = { \Hy@naturalnames false}
1454   ,naturalnames .default:n = {true}
1455   ,pageanchor .choice:
1456   ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1457   ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1458   ,pageanchor .default:n = {true}
1459   ,plainpages .choice:
1460   ,plainpages / true .code:n = { \Hy@plainpagestrue}
1461   ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1462   ,plainpages .default:n = {true}

```

```

1463 }
1464
1465 \keys_define:nn { hyp / setup }
1466 {
1467   ,linktoc .choices:nn = { none, section, all, page }
1468   {
1469     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1470   }
1471   ,linktoc / unknown .code:n =
1472   {
1473     \msg_warning:nnxxx { hyp } { unknown-choice }
1474     { linktoc }
1475     { none, section, all, page }
1476     { \exp_not:n {#1} }
1477   }
1478   ,linktocpage .choice:
1479   ,linktocpage / true .meta:n = {linktoc=page}
1480   ,linktocpage / false .meta:n = {linktoc=section}
1481   ,linktocpage .default:n = true
1482 }
1483

```

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

```

url (setup key) 1484 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1485 {
menu (setup key) 1486   \keys_define:nn { hyp / setup }
run (setup key) 1487   {
1488     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1489   }
1490 }

```

```

1491 \keys_define:nn { hyp / setup }
1492 {
1493   ,baseurl .code:n =
1494   {
1495     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1496     \tl_if_empty:NTF \l__hyp_tmpa_tl
1497     {
1498       \pdfmanagement_remove:nn {Catalog} { URI }
1499     }
1500     {
1501       \pdfmanagement_add:nnx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1502     }
1503   }
1504   %only false does something ...
1505   ,bookmarks .choice:
1506   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1507   ,bookmarks / true .code:n = {}
1508   ,bookmarks .default:n = {true}
1509   ,bookmarksnumbered .choice:
1510   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1511   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1512   ,bookmarksnumbered .default:n = {true}
1513   ,bookmarksopen .choice:

```

```

1514 ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1515 ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1516 ,bookmarksopen .default:n = {true}
1517 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1518 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1519 ,pdfcenterwindow .choice:
1520 ,pdfcenterwindow / false .code:n =
1521 {
1522   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1523 }
1524 ,pdfcenterwindow / true .code:n =
1525 {
1526   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1527 }
1528 ,pdfcenterwindow / .code:n =
1529 {
1530   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1531 }
1532 ,pdfcenterwindow / unknown .code:n =
1533 {
1534   \msg_warning:nnxx { hyp } { no-bool }
1535   { pdfcenterwindow }
1536   { \exp_not:n {#1} }
1537 }
1538 ,pdfcenterwindow .default:n = true
1539 ,pdfdirection .choice:
1540 ,pdfdirection / L2R .code:n =
1541 {
1542   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1543 }
1544 ,pdfdirection / R2L .code:n =
1545 {
1546   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1547 }
1548 ,pdfdirection / .code:n =
1549 {
1550   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1551 }
1552 ,pdfdirection / unknown .code:n =
1553 {
1554   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1555   { pdfdirection }
1556   { L2R , R2L }
1557   { \exp_not:n {#1} }
1558 }
1559 ,pdfdisplaydoctitle .choice:
1560 ,pdfdisplaydoctitle / false .code:n =
1561 {
1562   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1563 }
1564 ,pdfdisplaydoctitle / true .code:n =
1565 {
1566   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1567 }

```

```

1568 ,pdffdisplaydoctitle .default:n = true
1569 ,pdfduplex .choices:nn =
1570 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1571 {
1572   \pdf_version_compare:NnTF > {1.6}
1573   {
1574     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1575     { PrintDuplex } { /#1 }
1576   }
1577   {
1578     \msg_warning:nnxx
1579     {hyp}
1580     {ignore-deprecated-or-unknown-option-in-pdf-version}
1581     {pdfduplex}
1582     {\pdf_version:}
1583   }
1584 }%
1585 ,pdfduplex / .code:n =
1586 {
1587   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1588 }
1589 ,pdfduplex / unknown .code:n =
1590 {
1591   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1592   { pdfduplex }
1593   { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1594   { \exp_not:n {#1} }
1595 }
1596 ,pdffitwindow .choice:
1597 ,pdffitwindow / false .code:n =
1598 {
1599   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1600 }
1601 ,pdffitwindow / true .code:n =
1602 {
1603   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1604 }
1605 ,pdffitwindow / .code:n =
1606 {
1607   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1608 }
1609 ,pdffitwindow .default:n = true
1610 ,pdffitwindow / unknown .code:n =
1611 {
1612   \msg_warning:nnxx { hyp } { no-bool }
1613   { pdffitwindow }
1614   { \exp_not:n {#1} }
1615 }
1616 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1617 ,pdflinkmargin .initial:n = {1pt}
1618 ,pdfmenubar .choice:
1619 ,pdfmenubar / true .code:n =
1620 {
1621   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }

```

```

1622     }
1623 ,pdfmenubar / false .code:n =
1624     {
1625         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1626         { HideMenubar } { true }
1627     }
1628 ,pdfmenubar / .code:n =
1629     {
1630         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1631     }
1632 ,pdfmenubar .default:n = true
1633 ,pdfmenubar / unknown .code:n =
1634     {
1635         \msg_warning:nnxx { hyp } { no-bool }
1636         { pdfmenubar }
1637         { \exp_not:n {#1} }
1638     }
1639 ,pdfnewwindow .choice:
1640 ,pdfnewwindow / true .code:n =
1641     {
1642         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1643         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1644     }
1645 ,pdfnewwindow / false .code:n =
1646     {
1647         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1648         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1649     }
1650 ,pdfnewwindow / .code:n =
1651     {
1652         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1653         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1654     }
1655 ,pdfnonfullscreenpagemode .choices:nn =
1656     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1657     {
1658         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1659         { NonFullScreenPageMode } {/#1}
1660     }
1661 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1662     {
1663         \pdf_version_compare:NnTF < {1.6}
1664         {
1665             %message
1666         }
1667         {
1668             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1669             {NonFullScreenPageMode}{/UseAttachments}
1670         }
1671     }
1672 ,pdfnonfullscreenpagemode / .code:n =
1673     {
1674         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1675     }

```

```

1676 ,pdfnonfullscreenpagemode / unknown .code:n =
1677 {
1678   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1679   { pdfnonfullscreenpagemode }
1680   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1681   { \exp_not:n {#1} }
1682 }
1683 ,pdfnumcopies .code:n =
1684 {
1685   \pdf_version_compare:NnTF > {1.6}
1686   {
1687     \tl_if_empty:nTF {#1}
1688     {
1689       \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1690     }
1691     {
1692       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1693       {NumCopies}{#1}
1694     }
1695   }
1696   {
1697     \msg_warning:nnxx
1698     {hyp}
1699     {ignore-deprecated-or-unknown-option-in-pdf-version}
1700     {pdfnumcopies}
1701     {\pdf_version:}
1702   }
1703 }
1704 ,pdfpagelayout .choices:nn =
1705 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1706 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1707 ,pdfpagelayout / .code:n =
1708 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1709 ,pdfpagelayout / unknown .code:n =
1710 {
1711   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1712   { pdfpagelayout }
1713   { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1714   { \exp_not:n {#1} }
1715 }
1716 ,pdfpagemode .choices:nn =
1717 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1718 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1719 ,pdfpagemode / UseAttachments .code:n =
1720 {
1721   \pdf_version_compare:NnTF > {1.5}
1722   {
1723     \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1724   }
1725   {
1726     \msg_warning:nnxx
1727     {hyp}
1728     {ignore-deprecated-or-unknown-value-in-pdf-version}
1729     {UseAttachments}

```

```

1730         {\pdf_version:}
1731     }
1732 }
1733 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1734 ,pdfpagemode / unknown .code:n =
1735 {
1736     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1737     { pdfpagemode }
1738     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1739     { \exp_not:n {#1} }
1740 }
1741 ,pdfpagescrop .code:n =
1742 {
1743     \tl_if_empty:nTF {#1} %or blank?
1744     {
1745         \pdfmanagement_remove:nn {Pages} { CropBox }
1746     }
1747     {
1748         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1749     }
1750 }
1751 ,pdfpicktraybypdfsize .choice:
1752 ,pdfpicktraybypdfsize / true .code:n =
1753 {
1754     \pdf_version_compare:NnTF > {1.6}
1755     {
1756         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1757         { PickTrayByPDFSize } { true }
1758     }
1759     {
1760         \msg_warning:nnxx
1761         {hyp}
1762         {ignore-deprecated-or-unknown-option-in-pdf-version}
1763         {pdfpicktraybypdfsize}
1764         {\pdf_version:}
1765     }
1766 }
1767 ,pdfpicktraybypdfsize / false .code:n =
1768 {
1769     \pdf_version_compare:NnTF > {1.6}
1770     {
1771         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1772         { PickTrayByPDFSize } { false }
1773     }
1774     {
1775         \msg_warning:nnxx
1776         {hyp}
1777         {ignore-deprecated-or-unknown-option-in-pdf-version}
1778         {pdfpicktraybypdfsize}
1779         {\pdf_version:}
1780     }
1781 }
1782 ,pdfpicktraybypdfsize / .code:n =
1783 {

```



```

1784     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1785 }
1786 ,pdfpicktraybypdfsize / unknown .code:n =
1787 {
1788     \msg_warning:nxxx { hyp } { no-bool }
1789     { picktraybypdfsize }
1790     { \exp_not:n {#1} }
1791 }
1792 ,pdfprintarea .choices:nn =
1793 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1794 {
1795     \pdf_version_compare:NnTF < {2.0}
1796     {
1797         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1798         { PrintArea } { /#1 }
1799     }
1800     {
1801         \msg_warning:nxxx
1802         {hyp}
1803         {ignore-deprecated-or-unknown-option-in-pdf-version}
1804         {pdfprintarea}
1805         {\pdf_version:}
1806     }
1807 }%
1808 ,pdfprintarea / .code:n =
1809 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1810 ,pdfprintarea / unknown .code:n =
1811 {
1812     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1813     { pdfprintarea }
1814     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1815     { \exp_not:n {#1} }
1816 }
1817 ,pdfprintclip .choices:nn =
1818 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1819 {
1820     \pdf_version_compare:NnTF < {2.0}
1821     {
1822         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1823         { PrintClip } { /#1 }
1824     }
1825     {
1826         \msg_warning:nxxx
1827         {hyp}
1828         {ignore-deprecated-or-unknown-option-in-pdf-version}
1829         {pdfprintclip}
1830         {\pdf_version:}
1831     }
1832 }%
1833 ,pdfprintclip / .code:n =
1834 {
1835     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1836 }
1837 ,pdfprintclip / unknown .code:n =

```

```

1838 {
1839     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1840     { pdfprintclip }
1841     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1842     { \exp_not:n {#1} }
1843 }
1844 ,pdfprintpagerange .code:n =
1845 {
1846     \pdf_version_compare:NnTF > {1.6}
1847     {
1848         \tl_if_empty:nTF { #1}
1849         {
1850             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1851             { PrintPageRange }
1852         }
1853         {
1854             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1855             {PrintPageRange}{[#1]}
1856         }
1857     }
1858     {
1859         \msg_warning:nnxx
1860         {hyp}
1861         {ignore-deprecated-or-unknown-option-in-pdf-version}
1862         {pdfprintpagerange}
1863         {\pdf_version:}
1864     }
1865 }
1866 ,pdfprintscaling .choices:nn =
1867 { None, AppDefault }
1868 {
1869     \pdf_version_compare:NnTF > {1.5}
1870     {
1871         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1872         { PrintScaling } { /#1 }
1873     }
1874     {
1875         \msg_warning:nnxx
1876         {hyp}
1877         {ignore-deprecated-or-unknown-option-in-pdf-version}
1878         {pdfprintscaling}
1879         {\pdf_version:}
1880     }
1881 }%
1882 ,pdfprintscaling / .code:n =
1883 {
1884     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1885 }
1886 ,pdfprintscaling / unknown .code:n =
1887 {
1888     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1889     { pdfprintarea }
1890     { None, AppDefault }
1891     { \exp_not:n {#1} }

```

```

1892     }
1893 ,pdfremotestartview .code:n =
1894 {
1895     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1896     \exp_args:NNV
1897     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1898     {
1899         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1900     }
1901     {
1902         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1903         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1904     }
1905 }
1906 ,pdfremotestartview .initial:n = {Fit}
1907 % pdfstartpage is special as it shares code with pdfstartview
1908 ,pdfstartpage .code:n =
1909 {
1910     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1911     \bool_if:nTF
1912     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1913     {
1914         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1915     }
1916     {
1917         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1918         {
1919             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1920         }
1921     }
1922 }
1923 ,pdfstartpage .initial:n =1
1924 ,pdfstartview .code:n =
1925 {
1926     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1927     \exp_args:NNV
1928     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1929     {
1930         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1931     }
1932     {
1933         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1934         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1935     }
1936     \bool_if:nTF
1937     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1938     {
1939         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1940     }
1941     {
1942         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1943         {
1944             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1945         }

```

```

1946     }
1947 }
1948 ,pdfstartview .initial:n = Fit
1949 ,pdftoolbar .choice:
1950 ,pdftoolbar / true .code:n =
1951 {
1952     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1953 }
1954 ,pdftoolbar / false .code:n =
1955 {
1956     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1957     { HideToolbar } { true }
1958 }
1959 ,pdftoolbar / true .code:n =
1960 {
1961     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1962 }
1963 ,pdftoolbar .default:n = true
1964 ,pdftoolbar / unknown .code:n =
1965 {
1966     \msg_warning:nxxx { hyp } { no-bool }
1967     { pdftoolbar }
1968     { \exp_not:n {#1} }
1969 }
1970 % pdfview see below.
1971 ,pdfviewarea .choices:nn =
1972 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1973 {
1974     \pdf_version_compare:NnTF < {2.0}
1975     {
1976         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1977         { ViewArea } { /#1 }
1978     }
1979     {
1980         \msg_warning:nxxx
1981         {hyp}
1982         {ignore-deprecated-or-unknown-option-in-pdf-version}
1983         {pdfviewarea}
1984         {\pdf_version:}
1985     }
1986 }%
1987 ,pdfviewarea / .code:n =
1988 {
1989     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1990 }
1991 ,pdfviewarea / unknown .code:n =
1992 {
1993     \msg_warning:nxxxx { hyp } { unknown-choice+empty }
1994     { pdfviewarea }
1995     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1996     { \exp_not:n {#1} }
1997 }
1998 ,pdfviewclip .choices:nn =
1999 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

2000 {
2001   \pdf_version_compare:NnTF < {2.0}
2002   {
2003     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2004     { ViewClip } { /#1 }
2005   }
2006   {
2007     \msg_warning:nnxx
2008     {hyp}
2009     {ignore-deprecated-or-unknown-option-in-pdf-version}
2010     {pdfviewclip}
2011     {\pdf_version:}
2012   }
2013 }%
2014 ,pdfviewclip / .code:n =
2015 {
2016   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2017 }
2018 ,pdfviewclip / unknown .code:n =
2019 {
2020   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
2021   { pdfviewclip }
2022   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2023   { \exp_not:n {#1} }
2024 }
2025 ,pdfwindowui .choice:
2026 ,pdfwindowui / true .code:n =
2027 {
2028   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2029 }
2030 ,pdfwindowui / false .code:n =
2031 {
2032   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2033   { HideWindowUI } { true }
2034 }
2035 ,pdfwindowui / .code:n =
2036 {
2037   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2038 }
2039 ,pdfwindowui / unknown .code:n =
2040 {
2041   \msg_warning:nnxx { hyp } { no-bool }
2042   { pdfwindowui }
2043   { \exp_not:n {#1} }
2044 }
2045 ,pdfwindowui .default:n = true
2046 }

```

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

```

2047 \keys_define:nn { hyp / setup }
2048 {
2049   ,pdfview .code:n =
2050   {
2051     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}

```

```

2052 \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2053 {
2054   { xyz }
2055   {
2056     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2057     {
2058       \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2059       \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2060       {
2061         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2062       }
2063       {
2064         \tl_set:Nx \l__hyp_dest_pdfview_tl
2065         {
2066           \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2067         }
2068       }
2069     }
2070     {
2071       \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2072     }
2073   }
2074   { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2075   { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2076   { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2077   { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2078   { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2079   { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2080   { fitr }
2081   {
2082     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2083     {
2084       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2085     }
2086     {
2087       %ensure 4 values ...
2088       \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2089       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2090       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2091       \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2092       \hbox_set_to_wd:Nnn \l__hyp_dest_box
2093       {
2094         \fp_eval:n
2095         {
2096           round
2097           (
2098             abs
2099             (
2100               \seq_item:Nn\l__hyp_tmpa_seq{4}
2101               -
2102               (\seq_item:Nn\l__hyp_tmpa_seq{2})
2103             ),
2104           3
2105         )

```

```

2106         }bp
2107     }{}
2108     \box_set_dp:Nn \l__hyp_dest_box
2109     {
2110         \fp_eval:n
2111         {
2112             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2113         }bp
2114     }
2115     \box_set_ht:Nn \l__hyp_dest_box
2116     {
2117         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2118     }
2119 }
2120 }
2121 }
2122 {
2123     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2124     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2125 }
2126 }
2127 ,pdfview .initial:n = {xyz}
2128 }

```

## 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.

```

2129 \keys_define:nn { hyp / setup }
2130 {
2131     ,pdflang .code:n =
2132     {
2133         \tl_if_empty:nTF { #1 }
2134         {
2135             \pdfmanagement_remove:nn {Catalog} { Lang }
2136         }
2137         {
2138             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2139         }
2140         \__hyp_store_metadata:nn {pdflang}{#1}
2141     }
2142 }

```

### 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*) 2143 \cs\_new\_protected:Npn \\_\_hyp\_setup\_info\_key:nn #1 #2

**pdfproducer** (*setup key*) 2144 {

**pdfkeywords** (*setup key*) 2145 \keys\_define:nn { hyp / setup }

```

2146 {
2147     pdf#1 .code:n =
2148     {
2149         \tl_if_blank:nTF {##1}
2150         {
2151             \str_case:nnF { #1 }
2152             {
2153                 {creator}
2154                 {
2155                     \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2156                     \pdfmanagement_add:nnx {Info}{Creator}{()}
2157                 }
2158                 {producer}
2159                 {
2160                     \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2161                     \pdfmanagement_add:nnx {Info}{Producer}{()}
2162                 }
2163             }
2164             {
2165                 \pdfmanagement_remove:nn {Info}{#2}
2166             }
2167         }
2168         {
2169             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2170             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2171             {
2172                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2173             }
2174         }
2175         \__hyp_store_metadata:nn {pdf#1}{##1}
2176     }
2177 }
2178 \keys_define:nn { hyp / info }
2179 {
2180     #2 .code:n =
2181     {
2182         \tl_if_blank:nTF {##1}
2183         {
2184             \pdfmanagement_remove:nn {Info}{#2}
2185         }
2186         {
2187             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2188             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2189             {
2190                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2191             }
2192         }
2193         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2194     }
2195 ,unknown .code:n =
2196 {
2197     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2198     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2199     {

```



```

2200         \exp_args:Nno
2201         \pdfmanagement_add:nnx {Info}
2202         { \l_keys_key_str } {\l__hyp_tmpa_str}
2203     }
2204 }
2205 }
2206 }
2207 \__hyp_setup_info_key:nn {author} {Author}
2208 \__hyp_setup_info_key:nn {title} {Title}
2209 \__hyp_setup_info_key:nn {producer} {Producer}
2210 \__hyp_setup_info_key:nn {creator} {Creator}
2211 % ignored key: addtopdfcreator
2212 \__hyp_setup_info_key:nn {subject} {Subject}
2213 \__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.

```

2214 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2215 {
2216     \keys_define:nn { hyp / setup }
2217     {
2218         pdf#1 .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             \__hyp_store_metadata:nn {pdf#1}{##1}
2228         }
2229     }
2230     \keys_define:nn { hyp / info }
2231     {
2232         #2 .code:n =
2233         {
2234             \tl_if_blank:nTF {##1}
2235             {
2236                 \pdfmanagement_remove:nn {Info}{#2}
2237             }
2238             {
2239                 \pdfmanagement_add:nnx {Info}{#2}{(##1)}
2240             }
2241             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2242         }
2243     }
2244 }
2245
2246 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2247 \__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 ...

```

2248 \keys_define:nn { hyp / setup }

```

```

2249 {
2250   ,pdftrapped .code:n =
2251   {
2252     \exp_args:Nne
2253     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2254   }
2255   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2256   {
2257     \pdfmanagement_add:nnx {Info}{Trapped}
2258     {/
2259       \str_uppercase:f { \str_head:n { #1 } }
2260       \str_lowercase:f { \str_tail:n { #1 } }
2261     }
2262     \__hyp_store_metadata:nx {pdftrapped}
2263     {
2264       \str_uppercase:f { \str_head:n { #1 } }
2265       \str_lowercase:f { \str_tail:n { #1 } }
2266     }
2267   }
2268   ,_pdftrapped / unknown .code:n =
2269   {
2270     \msg_warning:nnxxx { hyp } { unknown-choice }
2271     { pdftrapped }
2272     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2273     { \exp_not:n {#1} }
2274   }
2275 }

```

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

```

2276 \keys_define:nn { hyp / setup }
2277 {
2278   pdfinfo .code:n =
2279   {
2280     \keys_set:nn { hyp / info } { #1 }
2281   }
2282 }

```

Now we set some default values

```

2283 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2284 \keys_set:nn { hyp / setup } {pdfauthor = }
2285 \keys_set:nn { hyp / setup } {pdftitle = }
2286 \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:

```

2287 \clist_map_inline:nn
2288 {
2289   ,pdfcopyright
2290   ,pdftype
2291   ,pdflicenseurl
2292   ,pdfauthortitle

```

```

2293 ,pdfcaptionwriter
2294 ,pdfmetalang
2295 ,pdfapart
2296 ,pdfaconformance
2297 ,pdfuapart
2298 ,pdfxstandard
2299 ,pdfsource
2300 ,pdfdocumentid
2301 ,pdfinstanceid
2302 ,pdfversionid
2303 ,pdfrendition
2304 ,pdfpublication
2305 ,pdfpubtype
2306 ,pdfbytes
2307 ,pdfnumpages
2308 ,pdfissn
2309 ,pdfeissn
2310 ,pdfisbn
2311 ,pdfbookedition
2312 ,pdfpublisher
2313 ,pdfvolumenum
2314 ,pdfissuenum
2315 ,pdfpagerange
2316 ,pdfdoi
2317 ,pdfurl
2318 ,pdfidentifier
2319 ,pdfsubtitle
2320 ,pdfpubstatus
2321 ,pdfcontactaddress
2322 ,pdfcontactcity
2323 ,pdfcontactregion
2324 ,pdfcontactpostcode
2325 ,pdfcontactcountry
2326 ,pdfcontactphone
2327 ,pdfcontactemail
2328 ,pdfcontacturl
2329 ,pdfdate
2330 }
2331 {
2332   \keys_define:nn { hyp / setup }
2333   {
2334     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2335   }
2336 }
2337
2338 \str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
2339 {
2340   \exp_args:Nnx
2341   \keys_set:nn {hyp / setup}
2342   {
2343     ,pdfapart = \pdfmeta_standard_item:n {level}
2344     ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
2345   }
2346 }

```

## 12.5 Transitions

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

```

2347 \keys_define:nn { hyp / setup }
2348 {
2349   pdfpageduration .code:n =
2350   {
2351     \tl_if_blank:nTF { #1 }
2352     {
2353       \pdfmanagement_remove:nn {Page}{Dur}
2354     }
2355     {
2356       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2357     }
2358   }
2359 }

```

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}}
```

The keys allowed in the argument of `pdfpagetransition` are

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

```

2360 \keys_define:nn { hyp / setup }
2361 {
2362   pdfpagetransition .code:n =
2363   {
2364     \tl_if_blank:nTF {#1}
2365     {
2366       \pdfmanagement_remove:nn {Page}{Trans}
2367     }

```

```

2368     {
2369         \group_begin:
2370         \keys_set:nn { hyp / trans }{style=R,#1}
2371         \pdf_object_unnamed_write:nx { dict }
2372         {
2373             \pdfdict_use:n {l__hyp_page/Trans}
2374         }
2375         \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2376         \group_end:
2377     }
2378 }
2379 }
2380 \keys_define:nn { hyp / trans }
2381 {
2382     ,style .choices:nn =
2383     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2384     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2385     ,style / unknown .code:n =
2386     {
2387         \msg_warning:nnxxx { hyp } { unknown-choice }
2388         { trans / style }
2389         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2390         { \exp_not:n {#1} }
2391     }
2392     ,duration .code:n =
2393     {
2394         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2395     }
2396     ,direction .choices:nn =
2397     {H,V}
2398     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2399     ,direction .choices:nn =
2400     {0,90,180,270,315}
2401     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2402     ,direction / None .code:n =
2403     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2404     ,direction / unknown .code:n =
2405     {
2406         \msg_warning:nnxxx { hyp } { unknown-choice }
2407         { trans / direction }
2408         {
2409             H~(horizontal,~only~Split,~Blinds),
2410             V~(vertical,~only~Split,~Blinds),
2411             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2412             90~(bottom~to~top,~only~Wipe),
2413             180~(right~to~left,~only~Wipe),
2414             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2415             315~(top~left~to~bottom,~only~Glitter),
2416             None~(only~Fly)
2417         }
2418         { \exp_not:n {#1} }
2419     }
2420     ,motion .choices:nn =
2421     {I,0}

```

```

2422 { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2423 ,motion / unknown .code:n =
2424 {
2425   \msg_warning:nnxxx { hyp } { unknown-choice }
2426   { trans / motion }
2427   { I~(inwards) , O~(outwards) }
2428   { \exp_not:n {#1} }
2429 }
2430 ,scale .code:n =
2431 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2432 ,opaque .choices:nn = {true,false}
2433 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2434 ,opaque / unknown .code:n =
2435 {
2436   \msg_warning:nnxxx { hyp } { unknown-choice }
2437   { trans / B }
2438   { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2439   { \exp_not:n {#1} }
2440 }
2441 % try to set unknown keys as style
2442 ,unknown .code:n =
2443 {
2444   % warning ...
2445   \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2446 }
2447 }

```

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

```

2448 \keys_set:known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2449 \NewDocumentCommand \MakeFieldObject { m m }
2450 {
2451   \pdfxform_new:nnn { #2 }{ } { #1 }
2452 }
2453
2454
2455 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2456 \prop_new:N \g__hyp_AcroForm_Fields_prop
2457
2458 \let\HyField@afields\ltx@empty
2459 \let\HyField@cofields\ltx@empty
2460 %% UF test for old pdftex removed
2461 %\let\HyField@AuxAddToFields\ltx@gobble
2462 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2463 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2464
2465 % the value doesn't matter, but with a prop we avoid duplicates and it is
2466 % clearly faster than removing them from a sequence
2467 \def\HyField@AuxAddToFields#1
2468 {
2469   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2470 }%
2471
2472 %fields with empty key get a value too -- lets hope that

```

```

2473 %this give the expected behaviour
2474 \def\HyField@AuxAddToCoFields #1 #2
2475 {
2476   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2477 }
2478
2479 \Hy@AtBeginDocument
2480 {
2481   \if@filesw
2482     \immediate\write\@mainaux{%
2483       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2484     }%
2485     \immediate\write\@mainaux{%
2486       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2487     }%
2488   \fi
2489   \let\HyField@AfterAuxOpen\@firstofone
2490 }%
2491
2492 \def\HyField@AddToFields
2493 {
2494   \exp_args:Nx\HyField__hypAddToFields
2495   {
2496     \pdfannot_box_ref_last:
2497   }
2498   \ifx\Fld@calculate@code\ltx@empty
2499   \else
2500     \begingroup
2501     \Hy@safe@activetrue
2502     \edef\Hy@temp{%
2503       \endgroup
2504       \if@filesw
2505         \write\@mainaux
2506         {
2507           \string\HyField@AuxAddToCoFields
2508           {
2509             \Fld@calculate@sortkey
2510           }
2511           {
2512             \pdfannot_box_ref_last:
2513           }
2514         }
2515       \fi
2516     }%
2517     \Hy@temp
2518   \fi
2519 }%
2520
2521 \def\HyField__hypAddToFields#1{
2522   \HyField@AfterAuxOpen{%
2523     \if@filesw
2524       \write\@mainaux{%
2525         \string\HyField@AuxAddToFields{#1}%
2526       }%

```

```

2527     \fi
2528   }%
2529 }%
2530
2531 \ExplSyntaxOff
2532 \ExplSyntaxOn
2533
2534 \def\@Form[#1]
2535 {
2536   \kvsetkeys{Form}{#1}
2537   \pdf@ifdraftmode{}
2538   {
2539     \Hy@FormObjects
2540     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2541     {
2542       \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2543       %\pdfmanagement_show:n { Catalog / AcroForm }
2544     }
2545     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2546     {
2547       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2548       {
2549         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2550       }
2551       \seq_sort:Nn \l__hyp_tmpa_seq
2552       {
2553         \str_compare:nNnTF {##1} > {##2}
2554         { \sort_return_swapped: }
2555         { \sort_return_same: }
2556       }
2557       \seq_map_inline:Nn \l__hyp_tmpa_seq
2558       {
2559         \pdfmanagement_add:nnx { Catalog / AcroForm }
2560         { CO }
2561         {
2562           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2563         }
2564       }
2565     }
2566     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2567     {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2568     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2569     {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2570     \pdfmanagement_add:nnx {Catalog /AcroForm}
2571     {DA}{(/Helv~10~Tf~0~g)}
2572     \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2573     {
2574       \legacy_if:nT { HyField@NeedAppearances }
2575       {
2576         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2577       }
2578     }
2579     {
2580       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}

```



```

2581     }
2582   }
2583 }
2584 \ExplSyntaxOff
2585 \let\@endForm\ltx@empty
2586 \let\HyAnn@AbsPageLabel\ltx@empty
2587 \let\Fld@pageobjref\ltx@empty
2588
2589 \ExplSyntaxOn
2590 \newcount\HyAnn@Count
2591 \HyAnn@Count=\ltx@zero
2592 \def\HyAnn@AbsPageLabel
2593 {
2594   \global\advance\HyAnn@Count by\ltx@one
2595   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2596   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}%
2597   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2598   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2599   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2600 }%
2601 \def\Fld@pageobjref
2602 {
2603   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2604   {
2605     /P~\pdf_pageobject_ref:n
2606     {
2607       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2608     }
2609   }
2610 }
2611 \ExplSyntaxOff
2612 \ExplSyntaxOn
2613 %% check if the attr should be set through
2614 %% hooks.
2615 %% check if options are missing.
2616 \def\@TextField[#1]#2{% parameters, label
2617   \def\Fld@name{#2}%
2618   \let\Fld@default\ltx@empty
2619   \let\Fld@value\@empty
2620   \def\Fld@width{\DefaultWidthofText}%
2621   \def\Fld@height{%
2622     \ifFld@multiline
2623       \DefaultHeightofTextMultiline
2624     \else
2625       \DefaultHeightofText
2626     \fi
2627   }%
2628   \begingroup
2629     \expandafter\HyField@SetKeys\expandafter{%
2630       \DefaultOptionsofText,#1%
2631     }%
2632     \PDFForm@Name
2633     \HyField@FlagsText
2634     \ifFld@hidden\def\Fld@width{1sp}\fi

```

```

2635 \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2636 \LayoutTextField{#2}{%
2637   \leavevmode
2638   \HyAnn@AbsPageLabel
2639   \Hy@escapeform\PDFForm@Text
2640   \pdfannot_box:nnnn
2641     {\Fld@width}
2642     {\Fld@height}
2643     {Opt} %is this correct?
2644     {\PDFForm@Text}
2645   \MakeTextField{\Fld@width}{\Fld@height}
2646   \HyField@AddToFields
2647 }%
2648 \endgroup
2649 }
2650 \providecommand\@curropt{}
2651 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2652   \def\Fld@name{#2}
2653   \let\Fld@default\relax
2654   \let\Fld@value\relax
2655   \def\Fld@width{\DefaultWidthofChoiceMenu}
2656   \def\Fld@height{\DefaultHeightofChoiceMenu}
2657   \begingroup
2658     \Fld@menulength=0 %
2659     \@tempdima\z@
2660     \clist_map_variable:nNn { #3 } \@curropt
2661     %\@for\@curropt:=#3\do
2662     {%
2663       \expandafter\Fld@checkequals\@curropt==\%
2664       \Hy@StepCount\Fld@menulength
2665       \settowidth{\@tempdimb}{\@currDisplay}%
2666       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2667     }%
2668     \advance\@tempdima by~15\p@
2669     \begingroup
2670       \HyField@SetKeys{#1}
2671     \edef\x{\endgroup
2672       \noexpand\expandafter
2673       \noexpand\HyField@SetKeys
2674       \noexpand\expandafter{%
2675         \expandafter\noexpand\curname DefaultOptionsof%
2676         \ifFld@radio
2677           Radio%
2678         \else
2679           \ifFld@combo
2680             \ifFld@popdown
2681               PopdownBox%
2682             \else
2683               ComboBox%
2684             \fi
2685           \else
2686             ListBox%
2687           \fi
2688         \fi

```

```

2689         \endcsname
2690     }%
2691 } \x
2692 \HyField@SetKeys{#1}%
2693 \PDFForm@Name
2694 \ifFld@hidden\def\Fld@width{1sp}\fi
2695 \ifx\Fld@value\relax
2696     \let\Fld@value\Fld@default
2697 \fi
2698 \LayoutChoiceField{#2}{%
2699     \ifFld@radio
2700         \HyField@FlagsRadioButton
2701         \__hypRadio{#3}%
2702     \else
2703         \begingroup
2704             \HyField@FlagsChoice
2705             \ifdim\Fld@width<\@tempdima
2706                 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2707                 \edef\Fld@width{\the\@tempdima}%
2708             \fi
2709             \ifFld@combo
2710             \else
2711                 \@tempdima=\the\Fld@menulength\Fld@charsize
2712                 \advance\@tempdima by~\Fld@borderwidth bp %
2713                 \advance\@tempdima by~\Fld@borderwidth bp %
2714                 \edef\Fld@height{\the\@tempdima}%
2715             \fi
2716             \__hypListbox{#3}%
2717         \endgroup
2718     \fi
2719 }%
2720 \endgroup
2721 }
2722
2723 \def\__hypRadio#1{%
2724     \Fld@listcount=0-%
2725     %\show\Fld@default
2726     \EdefEscapeName\Fld@default{\Fld@default}%
2727     \clist_map_variable:nNn { #1 } \@curropt
2728     %\@for\@curropt:=#1\do
2729     {%
2730         \expandafter\Fld@checkequals\@curropt==\%
2731         \EdefEscapeName\@currValue{\@currValue}%
2732         \Hy@StepCount\Fld@listcount
2733         \@currDisplay\space
2734         \leavevmode
2735         \HyAnn@AbsPageLabel
2736         \Hy@escapeform\PDFForm@Radio
2737         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2738         {
2739             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2740             {
2741                 \group_begin:
2742                 \fontfamily{pzd}

```

```

2743         \fontencoding{U}
2744         \fontseries{m}
2745         \fontshape{n}
2746         \selectfont
2747         \char123
2748         \group_end:
2749     }
2750 }
2751 \pdfannot_box:nnnx
2752 {\Fld@width}
2753 {\Fld@height}
2754 {Opt} %is this correct?
2755 {
2756     \PDFForm@Radio
2757     /AP
2758     <<
2759     /N
2760     <<
2761     /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2762     %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2763     >>
2764     >>
2765 }
2766 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2767 \int_compare:nNtT { \Fld@listcount } = { 1 }
2768 { \HyField@AddToFields }
2769 \c_space_tl % deliberate space between radio buttons
2770 % to do: --> should be configurable
2771 }%
2772 }
2773
2774 \newcount\Fld@listcount
2775 \def\__hypListbox#1
2776 {
2777     \HyField@PDFChoices{#1}
2778     \mode_leave_vertical:
2779     \HyAnn@AbsPageLabel
2780     \Hy@escapeform\PDFForm@List
2781     \pdf_link_user:nnn
2782         {widget} %perhaps we need more types??
2783         {\PDFForm@List}
2784         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2785     \HyField@AddToFields
2786 }
2787
2788
2789 \def\@PushButton[#1]#2{% parameters, label
2790     \def\Fld@name{#2}%
2791     \group_begin:
2792         \exp_args:No\HyField@SetKeys
2793         {
2794             \DefaultOptionsofPushButton,#1
2795         }
2796     \PDFForm@Name

```

```

2797 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2798 {
2799   \HyField@FlagsPushButton
2800   \legacy_if:nT {Fld@hidden}
2801   {
2802     \def\Fld@width{1sp}
2803   }
2804   \LayoutPushButtonField
2805   {
2806     \mode_leave_vertical:
2807     \HyAnn@AbsPageLabel
2808     \Hy@escapeform\PDFForm@Push
2809     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2810     \pdfannot_box:nnnn
2811     {\box_wd:N\l_tmpa_box}
2812     {\box_ht:N\l_tmpa_box}
2813     {\box_dp:N\l_tmpa_box} %is this correct?
2814     {\PDFForm@Push}
2815     {\box_use:N\l_tmpa_box}
2816     \HyField@AddToFields
2817   }
2818 }
2819 {
2820   \msg_error:nn { hyp }{ pdfa-no-push-button }
2821   \LayoutPushButtonField
2822   {
2823     \mode_leave_vertical:
2824     \MakeButtonField{#2}
2825   }
2826 }
2827 \group_end:
2828 }
2829
2830 \def\@Submit[#1]#2
2831 {
2832   \def\Fld@width {\DefaultWidthofSubmit}
2833   \def\Fld@height{\DefaultHeightofSubmit}
2834   \group_begin:
2835     \exp_args:No\HyField@SetKeys
2836     {
2837       \DefaultOptionsofSubmit,#1
2838     }
2839     \HyField@FlagsPushButton
2840     \HyField@FlagsSubmit
2841     \legacy_if:nT { Fld@hidden }
2842     {
2843       \def\Fld@width{1sp}
2844     }
2845     \mode_leave_vertical:
2846     \HyAnn@AbsPageLabel
2847     \Hy@escapeform\PDFForm@Submit
2848     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2849     \pdfxform_if_exist:nF
2850     { __hyp_xform_Submit }

```

```

2851 {
2852   \pdfxform_new:nnn { __hyp_xform_Submit }{}
2853   {
2854     \fbox{\color_select:n{yellow}\textsf{Submit}}
2855   }
2856   \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2857   {
2858     \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2859   }
2860 }
2861 \pdfannot_box:nnnn
2862 {\box_wd:N\l_tmpa_box}
2863 {\box_ht:N\l_tmpa_box}
2864 {\box_dp:N\l_tmpa_box} %is this correct?
2865 {
2866   \PDFForm@Submit
2867   /AP<<
2868     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2869     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2870   >>
2871 }
2872 \HyField@AddToFields
2873 \box_use:N\l_tmpa_box
2874
2875 \group_end:
2876 }
2877
2878 \def\@Reset[#1]#2
2879 {
2880   \def\Fld@width {\DefaultWidthofReset}
2881   \def\Fld@height{\DefaultHeightofReset}
2882   \group_begin:
2883     \exp_args:No\HyField@SetKeys
2884     {
2885       \DefaultOptionsofReset,#1
2886     }
2887   \mode_leave_vertical:
2888   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2889   {
2890     \HyField@FlagsPushButton
2891     \legacy_if:nT { Fld@hidden }
2892     { \def\Fld@width{1sp} }
2893     \HyAnn@AbsPageLabel
2894     \Hy@escapeform\PDFForm@Reset
2895     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2896     \pdfannot_box:nnnn
2897     {\box_wd:N\l_tmpa_box}
2898     {\box_ht:N\l_tmpa_box}
2899     {\box_dp:N\l_tmpa_box} %is this correct?
2900     { \PDFForm@Reset }
2901     \HyField@AddToFields
2902     \box_use:N \l_tmpa_box
2903   }
2904   {

```

```

2905         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2906         \MakeButtonField{#2}
2907     }
2908     \group_end:
2909 }
2910
2911 \def\@CheckBox[#1]#2
2912 {% parameters, label
2913     \def\Fld@name{#2}
2914     \def\Fld@default{0}
2915     \group_begin:
2916         \def\Fld@width {\DefaultWidthofCheckBox}
2917         \def\Fld@height{\DefaultHeightofCheckBox}
2918         \exp_args:No\HyField@SetKeys
2919         {
2920             \DefaultOptionsofCheckBox,#1
2921         }
2922         \PDFForm@Name
2923         \HyField@FlagsCheckBox
2924         \legacy_if:nT { Fld@hidden }
2925         {
2926             \def\Fld@width{1sp}
2927         }
2928     \LayoutCheckField{#2}
2929     {
2930         \mode_leave_vertical:
2931         \HyAnn@AbsPageLabel
2932         \Hy@escapeform\PDFForm@Check
2933         \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2934         {
2935             \pdfxform_new:nnn
2936             {__hyp_xform_CheckMarkYes}{-}
2937             {
2938                 \group_begin:
2939                 \fontfamily{pzd}
2940                 \fontencoding{U}
2941                 \fontseries{m}
2942                 \fontshape{n}
2943                 \selectfont
2944                 \char51
2945                 \group_end:
2946             }
2947             \pdfxform_new:nnn
2948             {__hyp_xform_CheckMarkOff}{-}
2949             {
2950                 \group_begin:
2951                 \fontfamily{pzd}
2952                 \fontencoding{U}
2953                 \fontseries{m}
2954                 \fontshape{n}
2955                 \selectfont
2956                 \phantom{\char51} %perhaps xetex needs some small glyph ..
2957                 \group_end:
2958             }

```

```

2959         }
2960         \pdfannot_box:nnnn
2961         {\Fld@width}
2962         {\Fld@height}
2963         {Opt} %is this correct?
2964         {\PDFForm@Check}
2965         \HyField@AddToFields %check if this works with xelatex ...
2966     }
2967     \group_end:
2968 }
2969 \ExplSyntaxOff
2970
2971 %hm. Should a luatex driver use type1 fonts in fields????
2972 \ExplSyntaxOn
2973 \def\Hy@FormObjects
2974 {
2975     \pdf_object_new:nn {__hyp/Encoding/pdfdoc } { dict }
2976     \pdf_object_new:nn {__hyp/Font/ZaDb } { dict }
2977     \pdf_object_new:nn {__hyp/Font/Helv } { dict }
2978     \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
2979     {
2980         /Type/Encoding
2981         /Differences[
2982             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2983             /ring/tilde
2984             \c_space_tl
2985             39/quotesingle
2986             \c_space_tl
2987             96/grave %
2988             \iow_newline:
2989             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2990             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2991             /quotedblbase/quotedblleft/quotedblright/quoteleft
2992             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2993             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2994             \iow_newline:
2995             164/currency
2996             \c_space_tl
2997             166/brokenbar
2998             \c_space_tl
2999             168/dieresis/copyright/ordfeminine
3000             \c_space_tl
3001             172/logicalnot/.notdef/registered/macron/degree/plusminus
3002             /twosuperior/threesuperior/acute/mu
3003             \c_space_tl
3004             183/periodcentered/cedilla/onesuperior/ordmasculine
3005             \c_space_tl
3006             188/onequarter/onehalf/threequarters
3007             \iow_newline:
3008             192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3009             /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3010             /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3011             /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3012             /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls

```



```

3013         /grave/aacute/acircumflex/atilde/adieresis/aring/ae
3014         /cedilla/egrave/eacute/ecircumflex/edieresis/igrave
3015         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3016         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3017         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3018     ]
3019 }
3020 \pdf_object_write:nn {__hyp/Font/ZaDb }
3021 {
3022     /Type/Font
3023     /Subtype/Type1
3024     /Name/ZaDb
3025     /BaseFont/ZapfDingbats
3026 }
3027 \pdf_object_write:nx {__hyp/Font/Helv }
3028 {
3029     /Type/Font
3030     /Subtype/Type1
3031     /Name/Helv
3032     /BaseFont/Helvetica
3033     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3034 }
3035 \global\let\Hy@FormObjects\relax
3036 }
3037 \ExplSyntaxOff
3038 \providecommand*{\Fld@pageobjref}{\}
3039 \ifcsname pdf@escapestring\endcsname
3040 \def\Hy@escapeform#1{%
3041     \ifHy@pdfescapeform
3042         \let\Hy@escapestring\pdfescapestring
3043     \else
3044         \let\Hy@escapestring\@firstofone
3045     \fi
3046 }%
3047 \Hy@escapeform{}\}
3048 \else
3049 \let\Hy@escapestring\@firstofone
3050 \def\Hy@escapeform#1{%
3051     \ifHy@pdfescapeform
3052         \def\Hy@escapestring##1{%
3053             \noexpand\Hy@escapestring{\noexpand##1}%
3054         }%
3055         \edef\Hy@temp{#1}%
3056         \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
3057         \def\Hy@escapestring##1{%
3058             \@ifundefined{Hy@esc@\string##1}{%
3059                 ##1%
3060                 \ThisShouldNotHappen
3061             }{%
3062                 \csname Hy@esc@\string##1\endcsname
3063             }%
3064         }%
3065     \else
3066         \let\Hy@escapestring\@firstofone

```

```

3067     \fi
3068   }%
3069   \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3070     \ifx\#3\%
3071     \else
3072       \expandafter
3073       \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3074       \ltx@ReturnAfterFi{%
3075         \Hy__hypescapeform#3\@nil
3076       }%
3077     \fi
3078   }%
3079   \fi
3080   \def\PDFForm@Name{%
3081     \PDFForm__hypName\Fld@name
3082     \ifx\Fld@altname\relax
3083     \else
3084       \PDFForm__hypName\Fld@altname
3085     \fi
3086     \ifx\Fld@mappingname\relax
3087     \else
3088       \PDFForm__hypName\Fld@mappingname
3089     \fi
3090   }
3091   \def\PDFForm__hypName#1{%
3092     \begingroup
3093     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3094       \ifHy@unicode
3095         \Hy@unicodedefalse
3096       \fi
3097     \fi
3098     \pdfstringdef\Hy@gtemp#1%
3099   \endgroup
3100   \let#1\Hy@gtemp
3101 }
3102 \def\Fld@X@additionalactions{%
3103   \ifx\Fld@keystroke@code\@empty
3104   \else
3105     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3106   \fi
3107   \ifx\Fld@format@code\@empty
3108   \else
3109     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3110   \fi
3111   \ifx\Fld@validate@code\@empty
3112   \else
3113     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3114   \fi
3115   \ifx\Fld@calculate@code\@empty
3116   \else
3117     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3118   \fi
3119   \ifx\Fld@onfocus@code\@empty
3120   \else

```

```

3121     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3122 \fi
3123 \ifx\Fld@onblur@code\@empty
3124 \else
3125     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3126 \fi
3127 \ifx\Fld@onmousedown@code\@empty
3128 \else
3129     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3130 \fi
3131 \ifx\Fld@onmouseup@code\@empty
3132 \else
3133     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3134 \fi
3135 \ifx\Fld@onenter@code\@empty
3136 \else
3137     /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3138 \fi
3139 \ifx\Fld@onexit@code\@empty
3140 \else
3141     /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3142 \fi
3143 }
3144 \ExplSyntaxOn
3145 \def\Fld@additionalactions
3146 {%
3147     \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3148     {
3149         \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3150         {/AA<<\Fld@X@additionalactions>>}
3151     }
3152 }
3153 \ExplSyntaxOff
3154 \def\Fld@annotnames{%
3155     /T(\Fld@name)%
3156     \ifx\Fld@altname\relax
3157     \else
3158         /TU(\Fld@altname)%
3159     \fi
3160     \ifx\Fld@mappingname\relax
3161     \else
3162         /TM(\Fld@mappingname)%
3163     \fi
3164 }
3165 \ExplSyntaxOn
3166 \def\PDFForm@Check
3167 {
3168     /Subtype/Widget
3169     ~\Fld@annotflags
3170     ~\Fld@pageobjref
3171     ~\Fld@annotnames
3172     /FT/Btn
3173     \Fld@flags
3174     /Q~\Fld@align

```

```

3175 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3176 /AP
3177 <<
3178 /N
3179 <<
3180 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3181 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3182 >>
3183 >>
3184 /MK<<
3185 \int_compare:nNf {\Fld@rotation}={0}
3186 {
3187 /R~\Fld@rotation
3188 }
3189 \tl_if_empty:NF\Fld@bordercolor
3190 {
3191 /BC[\Fld@bordercolor]
3192 }
3193 \tl_if_empty:NF\Fld@bcolor
3194 {
3195 /BG[\Fld@bcolor]
3196 }
3197 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3198 >>
3199 /DA
3200 (
3201 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3202 \tl_if_empty:NF \Fld@color
3203 {
3204 \c_space_tl \Fld@color
3205 }
3206 )
3207 /H/P
3208 \legacy_if:nTF {\Fld@checked}
3209 {
3210 /V/Yes /AS/Yes
3211 }
3212 {
3213 /V/Off /AS/Off
3214 }
3215 \Fld@additionalactions
3216 }
3217 \ExplSyntaxOff
3218 \ExplSyntaxOn
3219 \def\PDFForm@Push
3220 {
3221 /Subtype/Widget
3222 ~\Fld@annotflags
3223 ~\Fld@pageobjref
3224 ~\Fld@annotnames
3225 /FT/Btn
3226 ~\Fld@flags
3227 /H/P
3228 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>

```

```

3229     \bool_if:nT
3230     {
3231         !\int_compare_p:nNn {\Fld@rotation} = {0}
3232         ||
3233         \tl_if_exist_p:N \Fld@bordercolor
3234     }
3235     {
3236         /MK
3237         <<
3238         \int_compare:nNnF {\Fld@rotation} = {0}
3239         {
3240             /R~\Fld@rotation
3241         }
3242         \tl_if_exist:NT \Fld@bordercolor
3243         {
3244             /BC[\Fld@bordercolor]
3245         }
3246         >>
3247     }
3248     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3249     \Fld@additionalactions
3250 }
3251
3252 \ExplSyntaxOff
3253 \def\PDFForm@List{%
3254     /Subtype/Widget%
3255     \Fld@annotflags
3256     \Fld@pageobjref
3257     \Fld@annotnames
3258     /FT/Ch%
3259     \Fld@flags
3260     /Q \Fld@align
3261     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3262     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3263         \ifx\Fld@bordercolor\relax\else 1\fi
3264         \ifx\Fld@bcolor\relax \else 1\fi
3265         \space
3266     \else
3267         /MK<<%
3268         \ifnum\Fld@rotation=\z@
3269         \else
3270             /R \Fld@rotation
3271         \fi
3272         \ifx\Fld@bordercolor\relax
3273         \else
3274             /BC[\Fld@bordercolor]%
3275         \fi
3276         \ifx\Fld@bcolor\relax
3277         \else
3278             /BG[\Fld@bcolor]%
3279         \fi
3280         >>%
3281     \fi
3282     /DA(/Helv \strip@pt\Fld@charsize\space Tf%

```

```

3283     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3284 \Fld@choices
3285 \Fld@additionalactions
3286 }
3287 \ExplSyntaxOn
3288 \def\PDFForm@Radio
3289 {
3290   /Subtype/Widget
3291   ~\Fld@annotflags
3292   ~\Fld@pageobjref
3293   ~\Fld@annotnames
3294   /FT/Btn
3295   \Fld@flags
3296   /H/P
3297   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3298   /MK<<
3299   \ifnum\Fld@rotation=\z@
3300   \else
3301     /R~\Fld@rotation
3302   \fi
3303   \ifx\Fld@bordercolor\relax
3304   \else
3305     /BC[\Fld@bordercolor]%
3306   \fi
3307   \ifx\Fld@bcolor\relax
3308   \else
3309     /BG[\Fld@bcolor]%
3310   \fi
3311   /CA(\Hy@escapestring{\Fld@radiosymbol})%
3312   >>
3313   /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3314     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3315   \ifx\Fld@default\@empty
3316     /V/Off%
3317     /DV/Off%
3318   \else
3319     /V/\Fld@default
3320     /DV/\Fld@default
3321   \fi
3322   \Fld@additionalactions
3323 }
3324 \ExplSyntaxOff
3325 \ExplSyntaxOn
3326 % Does an appeareance dict make sense here?
3327 \def\PDFForm@Text
3328 {
3329   /Subtype/Widget
3330   ~\Fld@annotflags
3331   ~\Fld@pageobjref
3332   ~\Fld@annotnames
3333   /FT/Tx
3334   ~\Fld@flags
3335   /Q~\Fld@align
3336   /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>

```

```

3337 \bool_if:nT
3338 {
3339     !\int_compare_p:nNn {\Fld@rotation} = {0}
3340     ||
3341     \tl_if_exist_p:N \Fld@bordercolor
3342     ||
3343     \tl_if_exist_p:N \Fld@bcolor
3344 }
3345 {
3346     /MK
3347     <<
3348     \int_compare:nNf {\Fld@rotation} = {0}
3349     {
3350         /R~\Fld@rotation
3351     }
3352     \tl_if_exist:NT \Fld@bordercolor
3353     {
3354         /BC[\Fld@bordercolor]
3355     }
3356     \tl_if_exist:NT \Fld@bcolor
3357     {
3358         /BG[\Fld@bcolor]
3359     }
3360     >>
3361 }
3362 /DA
3363 (
3364     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3365     \tl_if_empty:NF {\c_space_tl\Fld@color}
3366 )
3367 /DV(\Hy@escapestring{\Fld@default})
3368 /V(\Hy@escapestring{\Fld@value})
3369 ~\Fld@additionalactions
3370 \int_compare:nNt { \Fld@maxlen}>{0}
3371 {
3372     /MaxLen~\Fld@maxlen
3373 }
3374 }
3375 \ExplSyntaxOff
3376
3377 \def\PDFForm@Submit{%
3378     /Subtype/Widget%
3379     \Fld@annotflags
3380     \Fld@pageobjref
3381     \Fld@annotnames
3382     /FT/Btn%
3383     \Fld@flags
3384     /H/P%
3385     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3386     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3387         \ifx\Fld@bordercolor\relax\else 1\fi
3388         \space
3389     \else
3390         /MK<<%

```

```

3391 \ifnum\Fld@rotation=\z@
3392 \else
3393 /R \Fld@rotation
3394 \fi
3395 \ifx\Fld@bordercolor\relax
3396 \else
3397 /BC[\Fld@bordercolor]%
3398 \fi
3399 >>%
3400 \fi
3401 /A<<%
3402 /S/SubmitForm%
3403 /F<<%
3404 /FS/URL%
3405 /F(\Hy@escapestring{\Form@action})%
3406 >>%
3407 \Fld@submitflags
3408 >>%
3409 \Fld@additionalactions
3410 }
3411 \ExplSyntaxOn
3412 \def\PDFForm@Reset{%
3413 /Subtype/Widget%
3414 \Fld@annotflags
3415 \Fld@pageobjref
3416 \Fld@annotnames
3417 /FT/Btn%
3418 \Fld@flags
3419 /H/P%
3420 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3421 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3422 \ifx\Fld@bordercolor\relax\else 1\fi
3423 \space
3424 \else
3425 /MK<<%
3426 \ifnum\Fld@rotation=\z@
3427 \else
3428 /R~\Fld@rotation
3429 \fi
3430 \ifx\Fld@bordercolor\relax
3431 \else
3432 /BC[\Fld@bordercolor]%
3433 \fi
3434 >>%
3435 \fi
3436 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3437 /A<</S/ResetForm>>%
3438 \Fld@additionalactions
3439 }%
3440
3441
3442 %these patterns are used in hyperref checks.
3443 %it is unclear if they are really useful and if a backend support is
3444 %needed.

```



```

3445 \str_case:VnF \c_sys_backend_str
3446 {
3447   { pdfmode }
3448   {
3449     \def\HyPat@ObjRef
3450     {
3451       [0-9]*[1-9][0-9]*~0~R
3452     }
3453   }
3454   { dvipdfmx }
3455   {
3456     \def\HyPat@ObjRef
3457     {
3458       @[\~]+
3459     }
3460   }
3461   { xdvipdfmx }
3462   {
3463     \def\HyPat@ObjRef
3464     {
3465       @[\~]+
3466     }
3467   }
3468 }
3469 { %also set in hyperref sty, so probably not needed.
3470   \def\HyPat@ObjRef/{.+}
3471 }
3472
3473
3474 \ExplSyntaxOff
3475 % UF: removed Hy@writebookmark
3476 %   \Hy@currentbookmarklevel{0}
3477 %   \Hy@numberline
3478 %   \__hypwritetorep
3479 %   counter{bookmark@seq@number}
3480 % removed \HyPsd@SanitizeForOutFile, not needed
3481 % removed \currentpdfbookmark, defined by bookmark,
3482 % should use \newcommand there
3483 % removed \subpdfbookmark, defined by bookmark,
3484 % should use \newcommand there
3485 % removed \belowpdfbookmark, defined by bookmark,
3486 % should use \newcommand there
3487 % removed \pdfbookmark, defined by bookmark,
3488 % \BOOKMARK
3489 % \@BOOKMARK
3490 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3491 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3492 %% removed \ReadBookmarks / unneeded with bookmark.
3493 %% removed \Hy@OutlineName
3494 %% removed \check@bm@number
3495 %% removed \calc@bm@number
3496
3497 \ifHy@implicit
3498 \else

```

```

3499 \expandafter\endinput
3500 \fi
3501 \newlength\Hy@SectionHShift
3502 \def\Hy@SectionAnchorHref#1{%
3503 \ifx\protect\@typeset@protect
3504 \Hy__hypSectionAnchor{#1}%
3505 \fi
3506 }
3507 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3508 \leavevmode
3509 \hbox to Opt{%
3510 \kern-\Hy@SectionHShift
3511 \Hy@raisedlink{%
3512 \hyper@anchorstart{#1}\hyper@anchorend
3513 }%
3514 \hss
3515 }%
3516 }
3517 \let\H@old@ssect\@ssect
3518 \def\@ssect#1#2#3#4#5{%
3519 \Hy@MakeCurrentHrefAuto{section*}%
3520 \setlength{\Hy@SectionHShift}{#1}%
3521 \begingroup
3522 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3523 \toks\tw@\expandafter{%
3524 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3525 #5%
3526 }%
3527 \edef\x{\endgroup
3528 \the\toks@\the\toks\tw@}%
3529 }\x
3530 }
3531 \let\H@old@schapter\@schapter
3532 \def\@schapter#1{%
3533 \begingroup
3534 \let\@mkboth\@gobbletwo
3535 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3536 \Hy@raisedlink{%
3537 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3538 }%
3539 \endgroup
3540 \H@old@schapter{#1}%
3541 }
3542 \ltx@ifundefined{chapter}{-}{%
3543 \let\Hy@org@chapter\@chapter
3544 \def\@chapter{%
3545 \def\Hy@next{%
3546 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3547 \Hy@raisedlink{%
3548 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3549 }%
3550 }%
3551 \ifnum\c@secnumdepth>\m@ne
3552 \ltx@ifundefined{if@mainmatter}%

```

```

3553     \iftrue{\csname if@mainmatter\endcsname}%
3554     \let\Hy@next\relax
3555     \fi
3556   \fi
3557   \Hy@next
3558   \Hy@org@chapter
3559 }%
3560 }
3561 \let\H@old@part\@part
3562 \begingroup\expandafter\expandafter\expandafter\endgroup
3563 \expandafter\ifx\csname chapter\endcsname\relax
3564   \let\Hy@secnum@part\z@
3565 \else
3566   \let\Hy@secnum@part\m@ne
3567 \fi
3568 \def\@part{%
3569   \ifnum\Hy@secnum@part>\c@secnumdepth
3570     \phantomsection
3571   \fi
3572   \H@old@part
3573 }
3574 \let\H@old@spart\@spart
3575 \def\@spart#1{%
3576   \Hy@MakeCurrentHrefAuto{part*}%
3577   \Hy@raisedlink{%
3578     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3579   }%
3580   \H@old@spart{#1}%
3581 }
3582 \let\H@old@sect\@sect
3583 \def\@sect#1#2#3#4#5#6[#7]#8{%
3584   \ifnum #2>\c@secnumdepth
3585     \expandafter\@firstoftwo
3586   \else
3587     \expandafter\@secondoftwo
3588   \fi
3589   {%
3590     \Hy@MakeCurrentHrefAuto{section*}%
3591     \setlength{\Hy@SectionHShift}{#3}%
3592     \begingroup
3593       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3594       \toks\tw@\expandafter{%
3595         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3596         #8%
3597       }%
3598       \edef\x{\endgroup
3599         \the\toks@{\the\toks\tw@}}%
3600       }\x
3601   }{%
3602     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3603   }%
3604 }
3605 \expandafter\def\csname Parent-4\endcsname{}
3606 \expandafter\def\csname Parent-3\endcsname{}

```

```

3607 \expandafter\def\csname Parent-2\endcsname{}
3608 \expandafter\def\csname Parent-1\endcsname{}
3609 \expandafter\def\csname Parent0\endcsname{}
3610 \expandafter\def\csname Parent1\endcsname{}
3611 \expandafter\def\csname Parent2\endcsname{}
3612 \expandafter\def\csname Parent3\endcsname{}
3613 \expandafter\def\csname Parent4\endcsname{}
3614 %%
3615 %% End of file 'hgeneric-testphase.def'.
3616 \package

3617 \*colorscheme
3618 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3619 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3620 % linkcolor=
3621 %,filecolor=
3622 %,urlcolor=
3623 %,menucolor=
3624 %,runcolor=
3625 %,linkbordercolor=
3626 %,filebordercolor=
3627 %,urlbordercolor=
3628 %,menubordercolor=
3629 %,runbordercolor=
3630
3631 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3632 {
3633     linkcolor      = [rgb]{1,0,0}, %red
3634     filecolor      = [rgb]{0,1,1}, %cyan
3635     urlcolor       = [rgb]{1,0,1}, %magenta
3636     menucolor      = [rgb]{1, 0, 0}, %red
3637     runcolor       = [rgb]{0,1,1}, %cyan
3638     %-----
3639     linkbordercolor = [rgb]{1, 0 ,0 },
3640     filebordercolor = [rgb]{0, .5, .5},
3641     urlbordercolor  = [rgb]{0, 1, 1},
3642     menubordercolor = [rgb]{1, 0, 0},
3643     runbordercolor  = [rgb]{0, .7, .7}
3644 }
3645
3646 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3647 {
3648     linkcolor      = [rgb]{0,0.2,0.6},
3649     filecolor      = [rgb]{0.8,0,0.8},
3650     urlcolor       = [rgb]{0.8,0,0.8},
3651     menucolor      = [rgb]{0,0.2,0.6},
3652     runcolor       = [rgb]{0.8,0,0.8},
3653     %----- %-----
3654     linkbordercolor = [rgb]{0,0.2,0.6},
3655     filebordercolor = [rgb]{0.8,0,0.8},
3656     urlbordercolor  = [rgb]{0.8,0,0.8},
3657     menubordercolor = [rgb]{0,0.2,0.6},
3658     runbordercolor  = [rgb]{0.8,0,0.8}
3659 }
3660

```

```

3661 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3662 { %two colors: intern/extern
3663   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3664   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3665   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3666   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3667   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3668   %----- %-----
3669   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3670   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3671   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3672   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3673   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3674 }
3675
3676 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3677 { %all darkgray
3678   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3679   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3680   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3681   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3682   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3683   %----- %-----
3684   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3685   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3686   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3687   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3688   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3689 }
3690
3691 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3692 { %dvipsnam.def
3693   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3694   filecolor      = [rgb]{1, 0, 0}, %Red
3695   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3696   menucolor      = [rgb]{1, 0, 0}, %Red
3697   runcolor       = [rgb]{1, 0, 0}, %Red
3698   %----- %-----
3699   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3700   filebordercolor = [rgb]{1, 0, 0}, %Red
3701   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3702   menubordercolor = [rgb]{1, 0, 0}, %Red
3703   runbordercolor  = [rgb]{1, 0, 0} %Red
3704 }
3705
3706 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3707 { %dvipsnam.def
3708   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3709   filecolor      = [rgb]{0, 1, 0}, %Green
3710   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3711   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3712   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3713   %----- %-----
3714   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed

```

```

3715     filebordercolor = [rgb]{0, 1, 0},      %Green
3716     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3717     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3718     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3719 }
3720
3721
3722 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3723 {
3724     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3725     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3726     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3727     menucolor      = [rgb]{0.44706, 0.45882, 0},
3728     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3729     %----- %-----
3730     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3731     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3732     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3733     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3734     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3735 }
3736
3737 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3738 {
3739     linkcolor      = [rgb]{0,0,0},
3740     filecolor      = [rgb]{0,0,0},
3741     urlcolor       = [rgb]{0,0,0},
3742     menucolor      = [rgb]{0,0,0},
3743     runcolor       = [rgb]{0,0,0},
3744     %----- %-----
3745     linkbordercolor = [rgb]{0,0,0},
3746     filebordercolor = [rgb]{0,0,0},
3747     urlbordercolor  = [rgb]{0,0,0},
3748     menubordercolor = [rgb]{0,0,0},
3749     runbordercolor  = [rgb]{0,0,0}
3750 }
3751 </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.

Symbols	
\#	279, 737
\\$	278
\%	738
\.	547, 551, 553
@curropt commands:	
\@curropt:	2661, 2728
\\	19, 20, 30, 31, 32, 40, 44, 54, 74, 81, 88, 95, 102, 117, 122, 123, 131, 132, 141, 142, 143, 144, 145, 152, 159, 280, 281, 867, 2663, 2730, 3070
\_	545, 547, 551, 553
A	
\A	545
\Acrobatmenu	18, 172
\addcontentsline	13
\AddToDocumentProperties	392
\AddToHook	430, 443
\AddToHookNext	197
\advance	2594, 2668, 2712, 2713
allcolors (hypersetup key)	980
\author	2
B	
\b	549
\begingroup	208, 277, 379, 2500, 2628, 2657, 2669, 2703, 3092, 3521, 3533, 3562, 3592
\belowpdfbookmark	3485
\bgroup	274, 379
\BOOKMARK	3488
bookmarkstyle (hypersetup key)	13
bool commands:	
\bool_if:N	267, 297, 359, 684, 700, 712, 725, 733, 763, 821, 878, 950, 960, 1204, 1215
\bool_if:nTF	833, 1911, 1936, 3229, 3337
\bool_lazy_and:nnTF	427
\bool_lazy_or:nnTF	1234, 1283
\bool_new:N	202, 203, 531, 535, 539
\bool_set_true:N	540
bordercolormodel (hypersetup key)	13, 1001
box commands:	
\box_dp:N	621, 2813, 2864, 2899
\box_ht:N	620, 2812, 2863, 2898
\box_new:N	480, 542
\box_set_dp:Nn	2108
\box_set_ht:Nn	2115
\box_use:N	1221, 2815, 2873, 2902
\box_use_drop:N	1226
\box_wd:N	619, 2811, 2862, 2897
\l_tmpa_box	2809, 2811, 2812, 2813, 2815, 2848, 2862, 2863, 2864, 2873, 2895, 2897, 2898, 2899, 2902
C	
\catcode	278, 279
\char	2747, 2944, 2956
\chardef	157
\cite	33
clist commands:	
\clist_map_function:nN	124, 133
\clist_map_inline:nn	2287
\clist_map_variable:nNn	2660, 2727
color commands:	
\color_export:nnN	38, 435, 909, 1022
\color_select:n	917, 953, 1225, 2854, 2858
\color_select:nn	923
\color_set:nn	2, 397, 934
\color_set:nnn	2, 396, 940
color names:	
hyp/annot/file	528
hyp/annot/link	528
hyp/annot/menu	528
hyp/annot/run	528
hyp/annot/url	528
colorfile (hypersetup key)	980
colorlink (hypersetup key)	980
colorlinks (hypersetup key)	966
colormenu (hypersetup key)	980
colorrn (hypersetup key)	980
colorscheme (hypersetup key)	1, 1392
colorurl (hypersetup key)	980
cs commands:	
\cs_generate_variant:Nn	155, 156, 394, 459, 474, 479, 583, 909, 926, 943
\cs_gset:Npn	1196
\cs_gset_eq:NN	413, 424
\cs_if_exist:N	12
\cs_if_exist_p:N	428
\cs_new:Npn	117, 475
\cs_new_protected:Npn	208, 306, 324, 342, 369, 389, 404, 415, 453, 462, 559, 563, 567, 571, 584,

586, 590, 634, 642, 651, 655, 677, 682, 698, 710, 723, 761, 819, 876, 910, 921, 927, 938, 1140, 2143, 2214	\DocumentMetadata . . . . .
\cs_set:Nn . . . . . 214, 256	. . . 3, 10, 14, 15, 18, 12, 20, 32, 41, 47
\cs_set_eq:NN . . . . .	draft (hypersetup key) . . . . . 1416
. . . . . 160, 165, 166, 737, 738, 1469	
\cs_set_protected:Npn . . . . .	<b>E</b>
. . . . . 158, 179, 600, 867	\edef 2502, 2671, 2707, 2714, 3055, 3527, 3598
\csname . . . . . 2675, 3062, 3073, 3553, 3563, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613	\EdefEscapeName . . . . . 2726, 2731
\currentpdfbookmark . . . . . 3481	\egroup . . . . . 382
	\else . . . . . 2499, 2624, 2678, 2682, 2685, 2702, 2710, 3043, 3048, 3065, 3071, 3083, 3087, 3104, 3108, 3112, 3116, 3120, 3124, 3128, 3132, 3136, 3140, 3157, 3161, 3262, 3263, 3264, 3266, 3269, 3273, 3277, 3283, 3300, 3304, 3308, 3314, 3318, 3386, 3387, 3389, 3392, 3396, 3421, 3422, 3424, 3427, 3431, 3498, 3565, 3586
<b>D</b>	\endcsname . . . . . 2689, 3039, 3062, 3073, 3553, 3563, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613
\d . . . . . 547, 551, 553	\endgroup . . . . . 283, 286, 380, 382, 2503, 2648, 2671, 2717, 2720, 3099, 3527, 3539, 3562, 3598
debug (hypersetup key) . . . . . 1416	\endinput . . . . . 3499
\DeclareRobustCommand . . . . .	exp commands:
. . . 261, 290, 314, 332, 352, 382, 3507	\exp_args:Ne . . . . . 3147
\def . . . . . 164, 379, 380, 382, 595, 2463, 2467, 2474, 2492, 2521, 2534, 2592, 2601, 2616, 2617, 2620, 2621, 2634, 2635, 2651, 2652, 2655, 2656, 2694, 2723, 2775, 2789, 2790, 2802, 2830, 2832, 2833, 2843, 2878, 2880, 2881, 2892, 2911, 2913, 2914, 2916, 2917, 2926, 2973, 3040, 3050, 3052, 3057, 3069, 3080, 3091, 3102, 3145, 3154, 3166, 3219, 3253, 3288, 3327, 3377, 3412, 3449, 3456, 3463, 3470, 3502, 3518, 3532, 3544, 3545, 3568, 3575, 3583, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613	\exp_args:Nne . . . . . 2252
\DefaultHeightofCheckBox . . . . . 2917	\exp_args:Nnno . . . . . 326, 344
\DefaultHeightofChoiceMenu . . . . . 2656	\exp_args:Nno . . . . . 308, 371, 2200
\DefaultHeightofReset . . . . . 2881	\exp_args:Nnox . . . . . 771
\DefaultHeightofSubmit . . . . . 2833	\exp_args:NNV . . . . . 1896, 1927
\DefaultHeightofText . . . . . 2625	\exp_args:NnV . . . . . 636, 645
\DefaultHeightofTextMultiline . . . . . 2623	\exp_args:Nnx . . . . . 2340, 2445
\DefaultOptionsofCheckBox . . . . . 2920	\exp_args:No . . . . .
\DefaultOptionsofPushButton . . . . . 2794	. . . . 669, 673, 2792, 2835, 2883, 2918
\DefaultOptionsofReset . . . . . 2885	\exp_args:NV . . . . . 617, 624
\DefaultOptionsofSubmit . . . . . 2837	\exp_args:Nx . . . . 767, 2193, 2241, 2494
\DefaultOptionsofText . . . . . 2630	\exp_not:N . . . . . 867
\DefaultWidthofCheckBox . . . . . 2916	\exp_not:n . . . . .
\DefaultWidthofChoiceMenu . . . . . 2655	. . . 1331, 1365, 1476, 1536, 1557, 1594, 1614, 1637, 1681, 1714, 1739, 1790, 1815, 1842, 1891, 1968, 1996, 2023, 2043, 2273, 2390, 2418, 2428, 2439
\DefaultWidthofReset . . . . . 2880	\expandafter . . . . . 280, 2629, 2663, 2672, 2674, 2675, 2730, 3056, 3072, 3499, 3523, 3524, 3562, 3563, 3585, 3587, 3594, 3595, 3605, 3606, 3607, 3608, 3609, 3610, 3611, 3612, 3613
\DefaultWidthofSubmit . . . . . 2832	\ExplSyntaxOff . . . . .
\DefaultWidthofText . . . . . 2620	. . . 9, 2531, 2584, 2611, 2969, 3037, 3153, 3217, 3252, 3324, 3375, 3474
destlabel (hypersetup key) . . . . . 13	\ExplSyntaxOn . . . . .
dim commands:	. . . 7, 10, 2532, 2589, 2612, 2972, 3144, 3165, 3218, 3287, 3325, 3411
\dim_to_decimal_in_bp:n . . . . . 11, 160	
\do . . . . . 2661, 2728	
\documentclass . . . . . 21, 33	



extension (hypersetup key) . . . . .	13, 1438	hidemenu (hypersetup key) . . . . .	1368
<b>F</b>			
\fbbox . . . . .	2766, 2854, 2858	hiderun (hypersetup key) . . . . .	1368
\fi . . . . .	2488,	hideurl (hypersetup key) . . . . .	1368
	2515, 2518, 2527, 2626, 2634, 2635,	hook commands:	
	2666, 2684, 2687, 2688, 2694, 2697,	\hook_gput_code:nnn . . . . .	
	2706, 2708, 2715, 2718, 3045, 3067,	. . . . .	406, 417, 946, 956, 1200, 1211
	3077, 3079, 3085, 3089, 3096, 3097,	\hook_new:n . . . . .	395, 558, 633
	3106, 3110, 3114, 3118, 3122, 3126,	\hook_new_pair:nn . . . . .	
	3130, 3134, 3138, 3142, 3159, 3163,	. . . . .	259, 288, 313, 331, 350
	3262, 3263, 3264, 3271, 3275, 3279,	\hook_use:n . . . . .	
	3281, 3283, 3302, 3306, 3310, 3314,	. . . . .	263, 284, 293, 310, 317, 328, 335,
	3321, 3386, 3387, 3394, 3398, 3400,		346, 355, 375, 575, 639, 648, 689, 705
	3421, 3422, 3429, 3433, 3435, 3500,	\href . . . . .	4, 5, 19, 261
	3505, 3555, 3556, 3567, 3571, 3588	\hreflaunch . . . . .	5
file (hypersetup key) . . . . .	10, 1484	\hrefpdf . . . . .	5, 19, 314, 387
file commands:		\hrefrun . . . . .	5, 332, 388
\file_input:n . . . . .	8	\hrefurl . . . . .	5, 19, 290, 386
fileborderstyle (hypersetup key) . . . . .	14, 1094	\hss . . . . .	3514
filecolor (hypersetup key) . . . . .	980	Hy internal commands:	
final (hypersetup key) . . . . .	1416	\Hy_hyescapeform . . . . .	3056, 3069, 3075
\fontencoding . . . . .	2743, 2940, 2952	\Hy_hypSectionAnchor . . . . .	3504, 3507
\fontfamily . . . . .	2742, 2939, 2951	HyField internal commands:	
\fontseries . . . . .	2744, 2941, 2953	\HyField_hypAddToFields . . . . .	2494, 2521
\fontshape . . . . .	2745, 2942, 2954	hyp commands:	
fp commands:		\l_hyp_annot_colorfile_bool . . . . .	529
\fp_eval:n . . . . .	2066, 2094, 2110	\l_hyp_annot_colorlink_bool . . . . .	529
<b>G</b>		\l_hyp_annot_colormenu_bool . . . . .	529
\gdef . . . . .	280, 281	\l_hyp_annot_colorrun_bool . . . . .	529
\GetDocumentProperties . . . . .	3	\l_hyp_annot_colorurl_bool . . . . .	529
\global . . . . .	2594, 3035	\l_hyp_annot_ocgcolorfile_bool . . . . .	533
group commands:		\l_hyp_annot_ocgcolorlink_bool . . . . .	533
\group_begin: . . . . .	264, 294, 318, 336,	\l_hyp_annot_ocgcolormenu_bool . . . . .	533
	356, 573, 687, 703, 727, 765, 823,	\l_hyp_annot_ocgcolorrun_bool . . . . .	533
	880, 952, 1207, 1224, 2369, 2741,	\l_hyp_annot_ocgcolorurl_bool . . . . .	533
	2791, 2834, 2882, 2915, 2938, 2950	\l_hyp_annot_ocgcolorurl_bool . . . . .	533
\group_end: . . . . .	309, 327, 345,	\l_hyp_current_dest_name_tl . . . . .	
	374, 580, 692, 715, 751, 811, 861,	. . . . .	632, 688, 704
	904, 962, 1227, 1230, 2376, 2748,	hyp internal commands:	
	2827, 2875, 2908, 2945, 2957, 2967	\g__hyp_AcroForm_CoFields_prop . . . . .	
<b>H</b>		. . . . .	2455, 2476, 2545, 2547, 2562
\hbox . . . . .	379, 3509	\g__hyp_AcroForm_Fields_prop . . . . .	
hbox commands:		. . . . .	2456, 2469, 2540
\hbox_overlap_right:n . . . . .	1221	\l__hyp_annot_GoTo_bool . . . . .	684, 700, 712
\hbox_set:Nn . . . . .	2809, 2848, 2895	\l__hyp_annot_GoTo_bool\l__-	
\hbox_set:Nw . . . . .	1208	_hyp_annot_URI_bool\l__-	
\hbox_set_end: . . . . .	1217	hyp_annot_GoToR_bool\l__-	
\hbox_set_to_wd:Nnn . . . . .	2092	hyp_annot_Named_bool\l__-	
hidefile (hypersetup key) . . . . .	1368	hyp_annot_Launch_bool . . . . .	537
hidelink (hypersetup key) . . . . .	1368	\l__hyp_annot_GoToR_bool . . . . .	763
hidelinks (hypersetup key) . . . . .	1368	\l__hyp_annot_Launch_bool . . . . .	821
		\l__hyp_annot_Named_bool . . . . .	878
		\c__hyp_annot_types_seq . . . . .	
		. . . . .	492, 529, 533, 980, 1281, 1379
		\l__hyp_annot_URI_bool . . . . .	725

\g__hyp_bordercolormodel_str . . .	\__hyp_href_url_aux:nn . . . . . 304, 306
..... 437, 528, 1005, 1024	\l__hyp_href_url_encode_bool . . .
\__hyp_citebordercolor_hook_-	..... 202, 213, 251, 267, 297, 359
init: . . . . . 402, 415, 424	\__hyp_href_url_format: . . . . .
\__hyp_citecolor_hook_init: . . . .	..... 208, 214, 256, 372
..... 400, 404, 413	\l__hyp_href_url_ismap_bool . . . .
\__hyp_clist_display:n . 117, 124, 133	..... 203, 227, 733
\__hyp_color_select:n 39, 910, 910, 926	\l__hyp_href_url_protocol_tl . . . .
\__hyp_color_select_aux:wn . . . . .	..... 204, 215, 254, 308, 373
..... 910, 914, 921	\__hyp_link_goto_begin:nw . . . . .
\__hyp_color_set:nn . . . . . 38, 39,	..... 655, 690, 706
400, 402, 434, 927, 927, 943, 985, 1021	\__hyp_link_goto_end: . 677, 691, 714
\__hyp_color_set_aux:nwn 927, 931, 938	\c__hyp_map_annot_hyp_prop . . . . 492
\c__hyp_colorscheme_daleif_prop 3646	\c__hyp_map_hyp_annot_prop . . . . .
\c__hyp_colorscheme_henryford_-	..... 492, 944,
prop . . . . . 3737	1008, 1048, 1075, 1083, 1094, 1121,
\c__hyp_colorscheme_julian_prop 3661	1129, 1198, 1306, 1342, 1352, 1484
\c__hyp_colorscheme_phelype_prop	\__hyp_ocg_init: 1140, 1140, 1196, 1206
..... 3722	l__hyp_page/Trans . . . . . 556
\c__hyp_colorscheme_szabolcsA_-	\__hyp_PageLabels_gpush: 590, 590, 603
prop . . . . . 3691	\l__hyp_para_tmpa_tl 488, 832, 835, 844
\c__hyp_colorscheme_szabolcsB_-	\l__hyp_para_tmpa_tl\l__hyp_-
prop . . . . . 3706	text_tmpa_str\g__hyp_text_-
\c__hyp_colorscheme_tivv_prop . 3676	tmpa_str . . . . . 485
\l__hyp_dest_box . . . . . 31,	\__hyp_ref_check:nn 452, 462, 474, 2599
542, 619, 620, 621, 2092, 2108, 2115	\__hyp_ref_if_exist:nn . 452, 460, 461
\l__hyp_dest_name_tmpa_tl . . 485,	\__hyp_ref_if_exist:nnTF . . 464, 2603
658, 659, 665, 669, 671, 674, 784, 797	\__hyp_ref_label:nn 452, 453, 459, 2598
\l__hyp_dest_pdfremotestartview_-	\__hyp_ref_value:nn 452, 475, 479, 2607
tl . . . . . 511, 792, 1899, 1903	\__hyp_secondoftwoewithopt:wnn . . .
\g__hyp_dest_pdfstartpage_tl . . .	..... 385, 386, 387, 388
... 511, 1910, 1912, 1919, 1937, 1944	\__hyp_setup_info_date_key:nn . . .
\g__hyp_dest_pdfstartview_tl . . .	..... 2214, 2246, 2247
511, 1912, 1919, 1930, 1934, 1937, 1944	\__hyp_setup_info_key:nn . . 2143,
\l__hyp_dest_pdfview_tl . . . . .	2207, 2208, 2209, 2210, 2212, 2213
..... 527, 637, 646,	\__hyp_store_metadata:nn . . . . .
2061, 2064, 2071, 2074, 2075, 2076,	..... 389, 394, 2140,
2077, 2078, 2079, 2084, 2088, 2124	2175, 2193, 2227, 2241, 2262, 2334
\c__hyp_dest_startview_regex . . .	\__hyp_text_cleanup:N . 563, 563, 577
..... 543, 1897, 1928	\l__hyp_text_enc_dest_print_tl . . .
\c__hyp_dest_undefined_tl . . . . .	..... 514, 783
..... 491, 664, 665	\l__hyp_text_enc_dest_tl . . . . .
\__hyp_destination:nn . . . . .	..... 31, 514, 613, 670
..... 31, 606, 606, 637, 646	\l__hyp_text_enc_file_print_tl . . .
\l__hyp_filename_tmpa_tl . . . 485,	..... 514, 826
766, 768, 774, 775, 780, 827, 828, 844	\l__hyp_text_enc_info_print_tl . . .
\__hyp_href_pdf_aux:nn . . . . . 321, 324	..... 514, 588
\l__hyp_href_pdf_destination_tl . . .	\l__hyp_text_enc_para_print_tl . . .
..... 205, 216, 255, 326	..... 514, 831
\l__hyp_href_pdf_page_tl 206, 224, 791	\l__hyp_text_enc_uri_print_tl . . .
\__hyp_href_run_aux:nn . . . . . 339, 342	..... 269,
\l__hyp_href_run_parameter_tl . . .	272, 299, 302, 361, 364, 514, 730, 1495
..... 207, 228, 344	
\__hyp_href_url_aux:n . . . . . 366, 369	

\_hyp\_text\_pdfstring:nnN	.....	hidelink	.....	1368
.....	571, 571, 583, 585,	hidelinks	.....	1368
588, 611, 668, 728, 781, 824, 829, 1495		hidemenu	.....	1368
\_hyp\_text\_pdfstring\_info:nN	...	hiderun	.....	1368
.....	586, 586, 2169, 2187, 2197	hideurl	.....	1368
\_hyp\_text\_purify:nN	. 559, 559, 576	hypertextnames	.....	1438
\_hyp\_text\_string\_from\_unicode:nN		link	.....	10, 1484
.....	567, 567, 578	linkborder	.....	13
\g\_hyp\_text\_tmpa\_str	. 490, 579, 581	linkborderstyle	.....	14, 1094
\l\_hyp\_text\_tmpa\_str	.....	linkcolor	.....	980
.....	489, 576, 577, 578, 579	linkfileprefix	.....	1438
\l\_hyp\_tmpa\_box	480, 1208, 1221, 1226	linktoc	.....	1438
\l\_hyp\_tmpa\_int	.....	linktocpage	.....	1438
\l\_hyp\_tmpa\_seq	.....	localanchorname	.....	1438
.....	480, 1897,	menu	.....	10, 1484
1899, 1928, 1930, 2051, 2052, 2056,		menuborder	.....	13
2058, 2082, 2089, 2090, 2091, 2100,		menuborderstyle	.....	14, 1094
2102, 2112, 2117, 2549, 2551, 2557		menucolor	.....	980
\l\_hyp\_tmpa\_str	.....	naturalnames	.....	1438
.....	480, 2169, 2170, 2172,	nesting	.....	13
2187, 2188, 2190, 2197, 2198, 2202		ocgcolorfile	.....	1234
\l\_hyp\_tmpa\_tl	480, 614, 618, 626,	ocgcolorlink	.....	1234
1025, 1029, 1495, 1496, 1501, 1895,		ocgcolorlinks	.....	1234
1897, 1926, 1928, 2058, 2059, 2066		ocgcolormenu	.....	1234
\l\_hyp\_uri\_tmpa\_tl	... 485, 731, 732	ocgcolorrun	.....	1234
hyp/anchor	.....	ocgcolorurl	.....	1234
hyp/annot/file (color name)	.....	pageanchor	.....	1438
hyp/annot/link (color name)	.....	pdfauthor	.....	2143
hyp/annot/menu (color name)	.....	pdfborder	.....	13
hyp/annot/run (color name)	.....	pdfborderstyle	.....	14, 1094
hyp/annot/url (color name)	.....	pdfcreationdate	.....	14, 2214
hyp/text/pdfstring	.....	pdfcreator	.....	2143
\hypercalcbp	.....	pdfencoding	.....	1406
.....	11, 18, 160	pdfinfo	.....	2276
\HyperDestNameFilter	.....	pdfkeywords	.....	2143
.....	13, 612, 669	pdflang	.....	14, 2129
\hypersetup	2, 3, 5, 9, 10, 13, 18, 66, 99, 179	pdflinkmargin	.....	14
\hypersetup keys:		pdfmoddate	.....	14, 2214
allcolors	.....	pdfproducer	.....	2143
.....	980	pdfremotestartview	.....	11
bookmarkstyle	.....	pdfstartview	.....	11
.....	13, 1001	pdfsubject	.....	2143
bordercolormodel	.....	pdftitle	.....	2143
colorfile	.....	pdftrapped	.....	2248
colorlink	.....	pdfversion	.....	1406
colorlinks	.....	pdfview	.....	11, 2047
colormenu	.....	plainpages	.....	1438
colormenu	.....	run	.....	10, 1484
colorrun	.....	runborder	.....	13
colorrun	.....	runborderstyle	.....	14, 1094
colorscheme	.....	runcolor	.....	980
.....	1, 1392	unicode	.....	1406
colorurl	.....	url	.....	10, 1484
.....	980	urlborder	.....	13
debug	.....			
.....	1416			
destlabel	.....			
.....	13			
draft	.....			
.....	1416			
extension	.....			
.....	13, 1438			
file	.....			
.....	10, 1484			
fileborderstyle	.....			
.....	14, 1094			
filecolor	.....			
.....	980			
final	.....			
.....	1416			
hidefile	.....			
.....	1368			

urlborderstyle	14, 1094		
urlcolor	980		
verbose	1416		
hypertextnames (hypersetup key)	1438		
hypListbox internal commands:			
\_hypListbox	2716, 2775		
hypRadio internal commands:			
\_hypRadio	2701, 2723		
hypwritetorep internal commands:			
\_hypwritetorep	3478		
<b>I</b>			
\ifcase	3262, 3386, 3421		
\ifcsname	3039		
\ifdim	2666, 2705, 2706		
\ifnum	3093, 3262, 3268, 3299, 3386, 3391, 3421, 3426, 3551, 3569, 3584		
\iftrue	3553		
\ifx	2498, 2635, 2695, 3070, 3082, 3086, 3103, 3107, 3111, 3115, 3119, 3123, 3127, 3131, 3135, 3139, 3156, 3160, 3263, 3264, 3272, 3276, 3283, 3303, 3307, 3314, 3315, 3387, 3395, 3422, 3430, 3503, 3563		
\immediate	2482, 2485		
int commands:			
\int_compare:nNnTF	2056, 2082, 2767, 3185, 3238, 3348, 3370		
\int_compare_p:nNn	3231, 3339		
\int_eval:n	790		
\int_max:nn	791		
\int_new:N	483		
iow commands:			
\iow_newline:	2988, 2994, 3007		
<b>K</b>			
\kern	3510		
keys commands:			
\keys_define:nn	185, 193, 211, 249, 398, 966, 982, 989, 1001, 1010, 1035, 1050, 1069, 1096, 1115, 1238, 1252, 1264, 1287, 1293, 1308, 1337, 1368, 1381, 1392, 1406, 1416, 1425, 1438, 1465, 1486, 1491, 2047, 2129, 2145, 2178, 2216, 2230, 2248, 2276, 2332, 2347, 2360, 2380		
\l_keys_key_str	190, 2202, 2445		
\keys_set:nn	182, 219, 238, 265, 295, 319, 337, 357, 408, 419, 1398, 1405, 2253, 2280, 2283, 2284, 2285, 2286, 2341, 2370, 2445		
\keys_set_known:nn	2448		
\kvsetkeys	181, 2536		
<b>L</b>			
\label	9, 13		
\LayoutCheckField	2928		
\LayoutChoiceField	2698		
\LayoutPushButtonField	2804, 2821		
\LayoutTextField	2636		
\leavevmode	2637, 2734, 3508		
legacy commands:			
\legacy_if:nTF	167, 441, 448, 598, 2574, 2800, 2841, 2891, 2924, 3208		
\let	266, 296, 320, 338, 358, 386, 387, 388, 695, 746, 753, 807, 813, 857, 863, 906, 2458, 2459, 2461, 2462, 2489, 2585, 2586, 2587, 2618, 2619, 2653, 2654, 2696, 3035, 3042, 3044, 3049, 3066, 3100, 3517, 3531, 3534, 3543, 3554, 3561, 3564, 3566, 3574, 3582		
link (hypersetup key)	10, 1484		
linkborder (hypersetup key)	13		
linkborderstyle (hypersetup key)	14, 1094		
linkcolor (hypersetup key)	980		
linkfileprefix (hypersetup key)	1438		
linktoc (hypersetup key)	1438		
linktocpage (hypersetup key)	1438		
localanchorname (hypersetup key)	1438		
<b>M</b>			
\MakeButtonField	2809, 2824, 2848, 2895, 2906		
\MakeChoiceField	2784		
\MakeFieldObject	2449		
\MakeRadioField	2766		
\MakeTextField	2645		
\mbox	1218		
menu (hypersetup key)	10, 1484		
menuborder (hypersetup key)	13		
menuborderstyle (hypersetup key)	14, 1094		
menucolor (hypersetup key)	980		
mode commands:			
\mode_if_horizontal:TF	608, 630		
\mode_leave_vertical:	262, 292, 316, 334, 354, 657, 740, 799, 846, 883, 2778, 2806, 2823, 2845, 2887, 2930		
msg commands:			
\msg_error:nn	429, 2820, 2905		
\msg_info:nnn	2155, 2160		
\msg_line_context:	82		
\g_msg_module_name_prop	11		
\msg_new:nnn	50, 57, 62, 66, 70, 77, 84, 91, 98, 105, 111, 118, 127, 137, 148		
\msg_new:nnnn	14, 25, 36		
\msg_warning:nn	169, 1412		
\msg_warning:nnn	178, 661, 901		

`\msg_warning:nnnn` ..... 1256, 1297, 1534, 1578,  
1612, 1635, 1697, 1726, 1760, 1775,  
1788, 1801, 1826, 1859, 1875, 1902,  
1933, 1966, 1980, 2007, 2041, 2123  
`\msg_warning:nnnnn` ..... 189,  
1328, 1362, 1473, 1554, 1591, 1678,  
1711, 1736, 1812, 1839, 1888, 1993,  
2020, 2270, 2387, 2406, 2425, 2436

## N

`naturalnames` (hypersetup key) ..... [1438](#)  
`nesting` (hypersetup key) ..... [13](#)  
`\newcommand` ..... 163, 3482, 3484, 3486  
`\newcount` ..... 2590, 2774  
`\NewDocumentCommand` ..... 2449  
`\NewExpandableDocumentCommand` .... 385  
`\newlength` ..... 3501  
`\noexpand` ... 2672, 2673, 2674, 2675, 3053  
`\nolinkurl` ..... 4

## O

`ocgcolorfile` (hypersetup key) ..... [1234](#)  
`ocgcolorlink` (hypersetup key) ..... [1234](#)  
`ocgcolorlinks` (hypersetup key) ..... [1234](#)  
`ocgcolormenu` (hypersetup key) ..... [1234](#)  
`ocgcolorrun` (hypersetup key) ..... [1234](#)  
`ocgcolorurl` (hypersetup key) ..... [1234](#)

## P

`pageanchor` (hypersetup key) ..... [1438](#)  
`\paperwidth` ..... 3  
`\PassOptionsToPackage` ... 450, 1430, 1435  
pdf commands:  
`\pdf_bdcobject:nn` ..... 1220, 1223  
`\pdf_destination:nn` .... 27, 155, 625  
`\pdf_destination:nnnn` ..... 618  
`\pdf_emc:` ..... 1222, 1228  
`\pdf_link_user:nnn` ..... 2781  
`\pdf_name_from_unicode_e:n` .....  
..... 244, 770, 885  
`\pdf_object_if_exist:nTF` ..... 768  
`\pdf_object_new:nn` ..... 1142,  
1143, 1144, 1145, 2975, 2976, 2977  
`\pdf_object_ref:n` ..... 156, 780,  
1148, 1150, 1172, 1173, 1176, 1180,  
1185, 1190, 1195, 2567, 2569, 3033  
`\pdf_object_ref_last:` .... 801, 2375  
`\pdf_object_unnamed_write:nn` ...  
..... 800, 2371  
`\pdf_object_write:nn` ..... 1146,  
1152, 1162, 1174, 2978, 3020, 3027  
`\pdf_pageobject_ref:n` 1919, 1944, 2605  
`\pdf_string_from_unicode:nnN` .. 569  
`\pdf_version:` .....  
... 1582, 1701, 1730, 1764, 1779,  
1805, 1830, 1863, 1879, 1984, 2011  
`\pdf_version_compare:NnTF` .....  
1572, 1663, 1685, 1721, 1754, 1769,  
1795, 1820, 1846, 1869, 1974, 2001  
`\pdf_version_compare_p:Nn` .....  
..... 837, 1235, 1284  
`\pdf_version_major:` .....  
..... 166, 1236, 1259, 1285, 1301  
`\pdf_version_minor:` .. 165, 1259, 1301  
pdfannot commands:  
`\pdfannot_box:nnnn` .....  
... 2640, 2751, 2810, 2861, 2896, 2960  
`\pdfannot_box_ref_last:` .. 2496, 2512  
`\pdfannot_dict_put:nnn` .....  
..... 741, 801, 847, 886, 1026,  
1061, 1085, 1107, 1131, 1313, 1344  
`\pdfannot_dict_remove:nn` .. 1016,  
1056, 1077, 1102, 1123, 1321, 1354  
`\pdfannot_link:nnn` . 742, 802, 848, 887  
`\pdfannot_link_goto_begin:nw` .. 674  
`\pdfannot_link_goto_end:` ..... 679  
`\pdfannot_link_margin:n` ..... 8, 1616  
`\c_pdfannot_link_types_seq` .... 537  
pdfauthor (hypersetup key) ..... [2143](#)  
`\pdfbookmark` ..... 3487  
pdfborder (hypersetup key) ..... [13](#)  
pdfborderstyle (hypersetup key) . 14, [1094](#)  
pdfcreationdate (hypersetup key) 14, [2214](#)  
pdfcreator (hypersetup key) ..... [2143](#)  
`\pdfdest` ..... 11  
pdfdict commands:  
`\pdfdict_new:n` 556, 719, 756, 815, 872  
`\pdfdict_put:nnn` .....  
... 236, 243, 557, 720, 721, 732,  
735, 757, 758, 770, 777, 787, 797,  
816, 817, 828, 843, 873, 874, 884,  
1642, 1643, 1647, 1648, 2384, 2394,  
2398, 2401, 2403, 2422, 2431, 2433  
`\pdfdict_remove:nn` 233, 840, 1652, 1653  
`\pdfdict_use:n` .....  
... 741, 800, 847, 853, 886, 891, 2373  
pdfencoding (hypersetup key) ..... [1406](#)  
`\pdfescapestring` ..... 3042  
pdffile commands:  
`\pdffile_embed_file:nnn` ..... 772  
PDFForm internal commands:  
`\PDFForm_hypName` .....  
..... 3081, 3084, 3088, 3091  
pdfinfo (hypersetup key) ..... [2276](#)  
pdfkeywords (hypersetup key) ..... [2143](#)  
pdflang (hypersetup key) ..... 14, [2129](#)  
pdflinkmargin (hypersetup key) ..... 14

pdfmanagement commands:	prop commands:
\pdfmanagement_add:nn ..... 1625	\prop_const_from_keyval:Nn ..... 494, 502, 3631, 3646, 3661, 3676, 3691, 3706, 3722, 3737
\pdfmanagement_add:nnn ..... 592, 1172, 1173, 1195, 1501, 1526, 1542, 1546, 1566, 1574, 1603, 1658, 1668, 1692, 1706, 1718, 1723, 1748, 1756, 1771, 1797, 1822, 1854, 1871, 1917, 1942, 1956, 1976, 2003, 2032, 2138, 2156, 2161, 2172, 2190, 2201, 2225, 2239, 2257, 2356, 2375, 2542, 2559, 2566, 2568, 2570, 2576	\prop_gput:Nnn ..... 11, 2469, 2476
\pdfmanagement_if_active_p: ... 428	\prop_if_empty:NTF ..... 2545
\pdfmanagement_remove:nn ..... ... 1498, 1522, 1530, 1550, 1562, 1587, 1599, 1607, 1621, 1630, 1674, 1689, 1708, 1745, 1784, 1809, 1835, 1850, 1884, 1914, 1939, 1952, 1961, 1989, 2016, 2028, 2037, 2135, 2165, 2184, 2222, 2236, 2353, 2366, 2580	\prop_item:Nn ..... 2562
\pdfmanagement_show:n ..... 2543	\prop_map_inline:Nn ..... ..... 944, 1008, 1048, 1075, 1083, 1094, 1121, 1129, 1198, 1306, 1342, 1352, 1396, 1484, 2540, 2547
pdfmeta commands:	\prop_new:N ..... 2455, 2456
\pdfmeta_standard_item:n ..... ..... 2338, 2343, 2344	\protect 266, 296, 320, 338, 358, 466, 695, 746, 753, 807, 813, 857, 863, 906, 3503
\pdfmeta_standard_verify:nnTF ... ..... 881, 2797, 2888	\providecommand ..... ..... 161, 162, 2483, 2486, 2650, 3038
\pdfmeta_standard_verify:nTF ... ..... 2572, 3149	\ProvidesFile ..... 3
pdfmoddate (hypersetup key) ..... 14, 2214	
pdfproducer (hypersetup key) ..... 2143	<b>R</b>
pdfremotestartview (hypersetup key) .. 11	\ReadBookmarks ..... 3492
pdfstartview (hypersetup key) ..... 11	ref commands:
\pdfstringdef ..... 8, 9, 22, 28, 3098	\ref_if_exist:nn ..... 460
\pdfstringdefDisableCommands ..... ..... 386, 387, 388	\ref_label:nn ..... 456
pdfsubject (hypersetup key) ..... 2143	\ref_value:nn ..... 25, 477
pdftitle (hypersetup key) ..... 2143	\refstepcounter ..... 12
pdftrapped (hypersetup key) ..... 2248	regex commands:
pdfversion (hypersetup key) ..... 1406	\regex_const:Nn ..... 543
pdfview (hypersetup key) ..... 11, 2047	\regex_extract_once:NnNTF 1897, 1928
pdfxform commands:	\relax ..... 695, 746, 753, 807, 813, 857, 863, 906, 2653, 2654, 2695, 3035, 3082, 3086, 3156, 3160, 3263, 3264, 3272, 3276, 3303, 3307, 3387, 3395, 3422, 3430, 3554, 3563
\pdfxform_if_exist:nTF ..... ..... 2737, 2849, 2933	\RemoveFromHook ..... 1506
\pdfxform_new:nnn ..... .. 2451, 2739, 2852, 2856, 2935, 2947	\RenewDocumentCommand ..... 172
\pdfxform_ref:n ..... .. 2761, 2762, 2868, 2869, 3180, 3181	\RequirePackage ..... 6, 31, 445, 3490
\phantom ..... 2956	run (hypersetup key) ..... 10, 1484
\phantomsection ..... 12, 19, 3570	runborder (hypersetup key) ..... 13
plainpages (hypersetup key) ..... 1438	runborderstyle (hypersetup key) . 14, 1094
prg commands:	runcolor (hypersetup key) ..... 980
\prg_do_nothing: ..... 413, 424	
\prg_generate_conditional_- variant:Nnn ..... 461	<b>S</b>
\prg_new_eq_conditional:NNn ... 460	\selectfont ..... 2746, 2943, 2955
	seq commands:
	\seq_const_from_clist:Nn ..... 492
	\seq_count:N ..... 2056, 2082
	\seq_get_right:NN ..... 2058
	\seq_item:Nn ..... 1899, 1930, 2052, 2100, 2102, 2112, 2117
	\seq_map_inline:Nn ..... . 529, 533, 537, 980, 1281, 1379, 2557
	\seq_new:N ..... 482
	\seq_put_right:Nn ..... ..... 2089, 2090, 2091, 2549



<code>\seq_set_split:Nnn</code> .....	2051	<code>\@currDisplay</code> .....	2665, 2733
<code>\seq_sort:Nn</code> .....	2551	<code>\@currValue</code> .....	2731, 2761
<code>\setbox</code> .....	379	<code>\@currentHref</code> .....	
<code>\setlength</code> .....	3520, 3591	.....	3524, 3537, 3548, 3578, 3595
<code>\setpdflinkmargin</code> .....	8	<code>\@curropt</code> ..	2650, 2660, 2663, 2727, 2730
<code>\settowidth</code> .....	2665	<code>\@empty</code> .....	2619, 2635, 3103,
<code>\show</code> .....	2725	3107, 3111, 3115, 3119, 3123, 3127,	
sort commands:		3131, 3135, 3139, 3283, 3314, 3315	
<code>\sort_return_same:</code> .....	2555	<code>\@endForm</code> .....	2585
<code>\sort_return_swapped:</code> .....	2554	<code>\@esphack</code> .....	457
<code>\space</code> .....	470, 2733, 3265, 3282,	<code>\@firstofone</code> ..	2489, 3044, 3049, 3066
3283, 3313, 3314, 3388, 3420, 3423		<code>\@firstoftwo</code> .....	3585
<code>\spacefactor</code> .....	608, 630	<code>\@for</code> .....	2661, 2728
<code>\stockwidth</code> .....	3	<code>\@gobbletwo</code> .....	3534
str commands:		<code>\@hyper@launch</code> .....	37, 867
<code>\c_colon_str</code> .....	867	<code>\@ifnextchar</code> .....	274
<code>\c_hash_str</code> .....	737	<code>\@ifundefined</code> .....	3058
<code>\c_percent_str</code> .....	738	<code>\@latex@warning</code> .....	467
<code>\str_case:nnTF</code> .....	2151, 3445	<code>\@mainaux</code> .....	2482, 2485, 2505, 2524
<code>\str_case_e:nnTF</code> .....	2052	<code>\@mkboth</code> .....	3534
<code>\str_compare:nNnTF</code> .....	2553	<code>\@nil</code> .....	3056, 3069, 3075
<code>\str_gset:Nn</code> .....	1005	<code>\@part</code> .....	3561, 3568
<code>\str_gset_eq:NN</code> .....	579	<code>\@pdfauthor</code> .....	23
<code>\str_head:n</code> .....	2259, 2264	<code>\@pdfborder</code> .....	161
<code>\str_if_eq:nnTF</code> .....		<code>\@pdfborderstyle</code> .....	162
... 615, 2170, 2188, 2198, 2338, 3147		<code>\@savsf</code> .....	608, 630
<code>\str_if_eq_p:nn</code> .....	835, 1236, 1285	<code>\@schapter</code> .....	3531, 3532
<code>\str_lowercase:n</code> .....		<code>\@secondoftwo</code> .....	3587
... 2052, 2193, 2241, 2260, 2265		<code>\@sect</code> .....	3582, 3583
<code>\str_new:N</code> .....	484, 489, 490, 528	<code>\@spart</code> .....	3574, 3575
<code>\str_set:Nn</code> .....	561	<code>\@ssect</code> .....	3517, 3518
<code>\str_set_convert:Nnnn</code> .....	8	<code>\@tempdima</code> ..	2659, 2666, 2668, 2705,
<code>\str_set_eq:NN</code> .....	581	2706, 2707, 2711, 2712, 2713, 2714	
<code>\str_tail:n</code> .....	2260, 2265	<code>\@tempdimb</code> .....	2665, 2666
<code>\str_uppercase:n</code> .....	2253, 2259, 2264	<code>\@typeset@protect</code> .....	3503
<code>\string</code> .....	2483,	<code>\@unexpandable@protect</code> .....	
2486, 2507, 2525, 3058, 3062, 3073		.....	266, 296, 320, 338, 358
<code>\subpdfbookmark</code> .....	3483	<code>\BKM@color</code> .....	438
sys commands:		<code>\c@secnumdepth</code> .....	3551, 3569, 3584
<code>\c_sys_backend_str</code> .....	3445	<code>\calc@bm@number</code> .....	3495
		<code>\check@bm@number</code> .....	3494
		<code>\define@key</code> .....	432
		<code>\Fld@additionalactions</code> ..	3145, 3215,
		3249, 3285, 3322, 3369, 3409, 3438	
		<code>\Fld@align</code> .....	3174, 3260, 3335
		<code>\Fld@altname</code> ..	3082, 3084, 3156, 3158
		<code>\Fld@annotflags</code> .....	3169,
		3222, 3255, 3291, 3330, 3379, 3414	
		<code>\Fld@annotnames</code> .....	3154, 3171,
		3224, 3257, 3293, 3332, 3381, 3416	
		<code>\Fld@bcolor</code> .....	3193, 3195, 3276,
		3278, 3307, 3309, 3343, 3356, 3358	
		<code>\fld@bcolor</code> .....	3264

## T

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

<code>\@BOOKMARK</code> .....	3489
<code>\@CheckBox</code> .....	2911
<code>\@ChoiceMenu</code> .....	2651
<code>\@Form</code> .....	2534
<code>\@PushButton</code> .....	2789
<code>\@Reset</code> .....	2878
<code>\@Submit</code> .....	2830
<code>\@TextField</code> .....	2616
<code>\@bookmarksopenlevel</code> .....	1517
<code>\@bsphack</code> .....	455
<code>\@chapter</code> .....	3543, 3544

\Fld@bordercolor .....	3189,	2707, 2752, 2766, 2784, 2802, 2832,
	3191, 3233, 3242, 3244, 3263, 3272,	2843, 2880, 2892, 2916, 2926, 2961
	3274, 3303, 3305, 3341, 3352, 3354,	
	3387, 3395, 3397, 3422, 3430, 3432	
\Fld@borderstyle .....	3175,	\Fld@X@additionalactions .....
	3228, 3261, 3297, 3336, 3385, 3436	..... 3102, 3147, 3150
\Fld@borderwidth .	2712, 2713, 3175,	\Form@action .....
	3228, 3261, 3297, 3336, 3385, 3436	..... 3405
\Fld@calculate@code .	2498, 3115, 3117	\G@refundefinedtrue .....
\Fld@calculate@sortkey .....	2509	..... 466
\Fld@cbsymbol .....	3197	\H@old@part .....
\Fld@charsize .....		..... 3561, 3572
..	2711, 3201, 3282, 3313, 3364, 3420	\H@old@schapter .....
\Fld@checkequals .....	2663, 2730	..... 3531, 3540
\Fld@choices .....	3284	\H@old@sect .....
\Fld@color	3202, 3204, 3283, 3314, 3365	..... 3582, 3593, 3602
\Fld@default .....		\H@old@spart .....
...	2618, 2635, 2653, 2696, 2725,	..... 3574, 3580
	2726, 2914, 3315, 3319, 3320, 3367	\H@old@ssect .....
\Fld@flags .....	3173,	..... 3517, 3522
	3226, 3259, 3295, 3334, 3383, 3418	\href@ .....
\Fld@format@code .....	3107, 3109	..... 274, 280
\Fld@height .....		\href@split .....
	2621, 2642, 2645, 2656, 2714, 2753,	..... 280, 281
	2766, 2784, 2833, 2881, 2917, 2962	\Hy@abspage .....
\Fld@keystroke@code .....	3103, 3105	..... 602
\Fld@listcount	2724, 2732, 2767, 2774	\Hy@activeanchorfalse .....
\Fld@mappingname	3086, 3088, 3160, 3162	..... 653
\Fld@maxlen .....	3370, 3372	\Hy@activeanchortrue .....
\Fld@menulength ....	2658, 2664, 2711	..... 644
\Fld@name .....		\Hy@AtBeginDocument .....
..	2617, 2652, 2790, 2913, 3081, 3155	..... 2463, 2479
\Fld@onblur@code .....	3123, 3125	\Hy@bookmarksnumberedfalse ...
\Fld@onclick@code .....	3248	..... 1510
\Fld@onenter@code .....	3135, 3137	\Hy@bookmarksnumberedtrue ...
\Fld@onexit@code .....	3139, 3141	..... 1511
\Fld@onfocus@code .....	3119, 3121	\Hy@bookmarksopenfalse .....
\Fld@onmousedown@code ...	3127, 3129	..... 1514
\Fld@onmouseup@code ....	3131, 3133	\Hy@bookmarksopentruetrue .....
\Fld@pageobjref .....		..... 1515
.....	2587, 2601, 3038, 3170,	\Hy@bookmarkstype .....
	3223, 3256, 3292, 3331, 3380, 3415	..... 1518
\Fld@radiosymbol .....	3311	\Hy@chapapp .....
\Fld@rotation .....	3185,	..... 3535, 3546
	3187, 3231, 3238, 3240, 3262, 3268,	\Hy@colorlink .....
	3270, 3299, 3301, 3339, 3348, 3350,	..... 34
	3386, 3391, 3393, 3421, 3426, 3428	\Hy@currentbookmarklevel .....
\Fld@submitflags .....	3407	..... 3476
\Fld@validate@code .....	3111, 3113	\Hy@DisableOption .....
\Fld@value .....		..... 171
..	2619, 2635, 2654, 2695, 2696, 3368	\Hy@drafttrue .....
\Fld@width .....	2620,	..... 1429
	2634, 2641, 2645, 2655, 2694, 2705,	\Hy@escapeform .....
		..... 2639, 2736, 2780, 2808,
		2847, 2894, 2932, 3040, 3047, 3050
		\Hy@escapestring .....
		..... 3042,
		3044, 3049, 3052, 3053, 3056, 3057,
		3066, 3069, 3105, 3109, 3113, 3117,
		3121, 3125, 3129, 3133, 3137, 3141,
		3197, 3248, 3311, 3367, 3368, 3405
		\Hy@finaltrue .....
		..... 1434
		\Hy@FormObjects ....
		..... 2539, 2973, 3035
		\Hy@gtemp .....
		..... 3098, 3100
		\Hy@href .....
		..... 274
		\Hy@href@nextactionraw .....
		..... 237
		\Hy@href@page .....
		..... 225
		\Hy@hypertextnamesfalse .....
		..... 1444
		\Hy@hypertextnamestrue .....
		..... 1443
		\Hy@linkfileprefix .....
		..... 1446
		\Hy@linktoc .....
		..... 1469
		\Hy@localanchornamefalse .....
		..... 1449
		\Hy@localanchornamestrue .....
		..... 1448
		\Hy@MakeCurrentHref .....
		..... 198
		\Hy@MakeCurrentHrefAuto .....
		..... 3519, 3535, 3546, 3576, 3590
		\Hy@naturalnamesfalse .....
		..... 1453
		\Hy@naturalnamestrue .....
		..... 1452
		\Hy@next .....
		..... 3545, 3554, 3557
		\Hy@numberline .....
		..... 164, 3477



\Hy@org@chapter	3543, 3558	\HyField@PDFChoices	2777
\Hy@OutlineName	3493	\HyField@SetKeys	2629, 2670, 2673, 2692, 2792, 2835, 2883, 2918
\Hy@OutlineRerunCheck	3491	\HyPat@ObjRef	3449, 3456, 3463, 3470
\Hy@pageanchorfalse	1457	\hyper@link	282
\Hy@pageanchortrue	1456	\hyper@anchor	632
\Hy@pdfmajorversion	166	\hyper@anchorend	632, 3512, 3537, 3548, 3578
\Hy@pdfminorversion	165	\hyper@anchorstart	632, 3512, 3537, 3548, 3578
\Hy@pdfstringtrue	29, 574	\hyper@link	33, 34, 682
\Hy@pdfversion	3093	\hyper@linkend	33, 34, 710
\Hy@plainpagesfalse	1461	\hyper@linkfile	326, 761
\Hy@plainpagetrue	1460	\hyper@linklaunch	36, 344, 819, 870
\Hy@pstringdef	584, 3073	\hyper@linknamed	38, 174, 876
\Hy@PutCatalog	590	\hyper@linkstart	33, 34, 698
\Hy@raisedlink	3511, 3536, 3547, 3577	\hyper@linkurl	308, 372, 723
\Hy@RestoreLastskip	629	\hyper@normalise	274, 304, 321, 339, 366, 379
\Hy@safe@activetrue	739, 2501	\HyPL@Labels	592, 602
\Hy@SaveLastskip	609	\HyPL@StorePageLabel	590
\Hy@secnum@part	3564, 3566, 3569	\HyPsd@SanitizeForOutFile	3480
\Hy@SectionAnchorHref	3502, 3524, 3595	\if@filesw	2481, 2504, 2523
\Hy@SectionHShift	3501, 3510, 3520, 3591	\ifFld@combo	2679, 2709
\Hy@StepCount	2664, 2732	\ifFld@hidden	2634, 2694
\Hy@temp	2502, 2517, 3055, 3056	\ifFld@multiline	2622
\Hy@unicodefalse	3095	\ifFld@popdown	2680
\Hy@VerboseAnchor	610	\ifFld@radio	2676, 2699
\Hy@verbosefalse	1420	\ifHy@implicit	3497
\Hy@VerboseLinkStart	686, 702	\ifHy@pdfescapeform	3041, 3051
\Hy@VerboseLinkStop	693, 716, 749, 809, 859, 897	\ifHy@unicode	3094
\Hy@verbosetrue	1419	\kv@set@family@handler	177
\Hy@VersionChecked	157	\ltx@empty	2458, 2459, 2498, 2585, 2586, 2587, 2618
\Hy@WrapperDef	606	\ltx@gobble	2461
\Hy@xspace@end	690, 748, 808, 858, 896	\ltx@gobbletwo	2462
\HyAnn@AbsPageLabel	2586, 2592, 2638, 2735, 2779, 2807, 2846, 2893, 2931	\ltx@ifUndefined	3542, 3552
\HyAnn@Count	2590, 2591, 2594, 2595, 2596, 2597, 2598, 2599, 2603, 2607	\ltx@one	2594
\HyField@AddToFields	2492, 2646, 2768, 2785, 2816, 2872, 2901, 2965	\ltx@ReturnAfterFi	3074
\HyField@afields	2458	\ltx@zero	2591
\HyField@AfterAuxOpen	2463, 2489, 2522	\m@ne	3551, 3566
\HyField@AuxAddToCoFields	2462, 2474, 2486, 2507	\OBJ@OCG@view	163
\HyField@AuxAddToFields	2461, 2467, 2483, 2525	\p@	2668
\HyField@cofields	2459	\pdf@ifdraftmode	2537
\HyField@FlagsCheckBox	2923	\PDF@SetupDoc	158
\HyField@FlagsChoice	2704	\PDFForm@Check	2932, 2964, 3166
\HyField@FlagsPushButton	2799, 2839, 2890	\PDFForm@List	2780, 2783, 3253
\HyField@FlagsRadioButton	2700	\PDFForm@Name	2632, 2693, 2796, 2922, 3080
\HyField@FlagsSubmit	2840	\PDFForm@Push	2808, 2814, 3219
\HyField@FlagsText	2633	\PDFForm@Radio	2736, 2756, 3288
		\PDFForm@Reset	2894, 2900, 3412
		\PDFForm@Submit	2847, 2866, 3377
		\PDFForm@Text	2639, 2644, 3327

