

# The `hyperref-generic` module

## A generic driver for `hyperref`

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

Version 0.96m, released 2024-09-29

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

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

The driver *requires* the new PDF management code, so documents wanting to use it should start like this (this requires L<sup>A</sup>T<sub>E</sub>X-2022-06-01 or newer):

```
\DocumentMetadata %loads the PDF management and activates it
{
  %% options
  %% e.g. pdf version, backend:
  % pdfversion=1.7,
  % backend = dvipdfmx
}
```

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

## 1 Avoiding transition problems

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

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

---

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

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

## 2 Bookmarks / outlines

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

## 3 “Metadata”

“Metadata”, information 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 are added by code from `l3pdfmeta`. (without the `pdfmanagement` the XMP-metadata can be added with packages like `pdfx` and `hyperxmp`).

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

The current handling of the metadata is problematic:

- External package like `hyperxmp` wants to access them too and for this had to patch a number of internal `hyperref` commands—which is a problem if the internal commands change (as happens with this new driver)
- `hyperref` (and also `hyperxmp`) tries to deduce some data 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}}
```

- It is possible to store titles in more than one language. If the value begins with an “optional argument” which represents a language tag, the value is taken as a comma list and split. The first value is used for the Info dictionary, the others are used in the XMP-metadata. Commas in a title must then be protected with braces:

```
pdftitle = {[en]English Title,[de] Deutscher Titel,[fr]{titre français, avec comm
```

- 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:

```
\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'`.

## 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 `cmymk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmymk` (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_colorrund_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorrund_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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` or `colorlinks=true` will as before disable the `pdfborder` (`colorlinks=false` will leave the `pdfborder` untouched), 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      urlbordercolor = [rgb]{1,1,0}
color expression urlbordercolor = red!50!blue
command          urlbordercolor = \mycolor
```

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

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

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

**link margin** The driver sets a default link margin—this is identical to `pdftex` and `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.
- `destlabel` (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  
`url` (*setup key*) new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally  
`file` (*setup key*) disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference  
`run` (*setup key*) without link (this is naturally also possible with `\ref*{abc}`). This disables also all  
`menu` (*setup key*) hooks of the link type, so the link is for example no longer colored. It also removes the  
implicit grouping of the content.

`nested-links` (*setup key*)

Links are sometimes nested. E.g. if a section heading contains a reference it can lead to nested links in the table of contents or if `\nameref` is used. That is not forbidden and normally work as expected: If the link area overlap normally the inner link is "on top" and chosen at a click. But it is not always actually wanted, so with the `nested-links` (a boolean key) it is possible to disable such nested links.

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

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

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

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

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

### 13.1 Names of destinations

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

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

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

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

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

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

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

## 14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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.

`nesting` (*setup key*)

`pdfborder` (*setup key*) This key set accept as value three numbers or three numbers and an array describing

`linkborder` (*setup key*) a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should

`urlborder` (*setup key*) according to the reference set round corners, but PDF viewer seem to ignore it. The

`runborder` (*setup key*) third number is the line width of the border. Settings done with `pdfborderstyle` should

`menuborder` (*setup key*) take precedence.

`pdfborderstyle` (*setup key*) The value of this key is the content of the BS dictionary. As an example

`linkborderstyle` (*setup key*) `/Type/Border /W 1 /S/U /D[3 2]`

<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])

`pdfcreationdate` (*setup key*) Setting these keys is normally not needed. If they are used the values of the first

`pdfmoddate` (*setup key*) two keys are stored directly in the Info dictionary for `/Creationdate` and `/ModDate`.

`pdfmetadate` (*setup key*) All three keys are used in XMP-metadata. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. `D:20200202111111+01'00'`.

`pdflinkmargin` (*setup key*) As described in the hyperref manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdfflatex and luatex work by page, with dviPDFmx the setting is global (and has to be done in the preamble).

`pdflang` (*setup key*) The key will work, but it is recommended to the set the language in `\DocumentMetadata` instead.

## File I

# hyperref-generic driver implementation

```

1 <*package>
2 <@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2024-09-29 v0.96m %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?

```

Temporary command definition, can be remove when hyperref is update too.

```

7 \long\def\Hy@ReturnAfterFi#1\fi{\fi#1}
8 \ExplSyntaxOn
9 \file_input:n {hyperref-colorschemes.def}
10 \ExplSyntaxOff

```

# 1 messages

Redirect the message name:

```
11 \ExplSyntaxOn
12 \prop_gput:Nnn \g_msg_module_name_prop { hyp }{ hyperref }
```

At first a message for the testing of the resource management

```
13 \cs_if_exist:NTF \DocumentMetadata
14 {
15   \msg_new:nnnn
16     { hyp }
17     { missing-resource-management }
18     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
19     {
20       Activate~it~with \\\
21       \tl_to_str:n{\DocumentMetadata{<options>}}\\
22       before~\tl_to_str:n{\documentclass}
23     }
24 }
25 {
26   \msg_new:nnnn
27     { hyp }
28     { missing-resource-management }
29     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
30     {
31       Activate~it~with \\\
32       \tl_to_str:n{\RequirePackage{pdfmanagement-testphase}}\\
33       \tl_to_str:n{\DocumentMetadata{<options>}}\\
34       before~\tl_to_str:n{\documentclass}
35     }
36 }
```

The pdfversion should be set in \DocumentMetadata

```
37 \msg_new:nnnn
38 { hyp }
39 { pdfversion-disabled }
40 {
41   This~hyperref~driver~ignores~the~pdfversion~key!\\
42   Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata
43 }
44 {
45   For~example:\\
46   \tl_to_str:n
47   {
48     \DocumentMetadata { pdfversion=1.7 }
49   }
50 }
```

A generic message for ignored keys.

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

```
57 }
```

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

```
58 \msg_new:nnn
59 { hyp }
60 { pdfa-no-push-button }
61 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
62
63 \msg_new:nnn
64 { hyp }
65 { pdfa-no-reset-button }
66 { PDF/A:~Reset~action~is~prohibited }
```

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

```
71 \msg_new:nnn
72 { hyp }
73 { empty-destination-name }
74 {
75   Empty-destination~name,\\
76   using~'#1'
77 }
```

A message if the destination check fails

```
78 \msg_new:nnn
79 { hyp }
80 { invalid-destination-value }
81 {
82   Invalid~value~'#1'~of~'#2'  \\
83   is~replaced~by~'Fit'~\msg_line_context:.
84 }
```

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

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

```

103     Value-`#1`~is~unknown~or~deprecated~in\\
104     pdf-version-#2. Value-`#3`~is used instead.
105 }

```

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:

```

106 \msg_new:nnn
107 { hyp }
108 { unknown-key }
109 {
110     unknown-key-#2-of-module-`#1`~set-to-`#3`.
111 }
112 \msg_new:nnn
113 { hyp }
114 { unknown-key-to-Hyp }
115 {
116     ignored-in-family-Hyp-unknown-key-#1.
117 }

```

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

```

118 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\\}
119 \msg_new:nnn
120 { hyp }
121 { unknown-choice }
122 {
123     Value-`#3`~is~invalid~for~key-`#1`.\\
124     The~key~accepts~only~the~choices\\
125     \clist_map_function:nN { #2 }\__hyp_clist_display:n
126 }
127
128 \msg_new:nnn
129 { hyp }
130 { unknown-choice+empty }
131 {
132     Value-`#3`~is~invalid~for~key-`#1`.\\
133     The~key~accepts~only~the~choices\\
134     \clist_map_function:nN { #2 }\__hyp_clist_display:n
135     An~empty~value~removes~the~setting.
136 }
137
138 \msg_new:nnn
139 { hyp }
140 { no-bool }
141 {
142     Value-`#2`~is~invalid~for~key-`#1`.\\
143     The~key~accepts~only~the~choices\\
144     *~true\\
145     *~false \\
146     *~and~an~empty~value~which~removes~the~setting.\\
147     No~value~is~equivalent~to~using~`true`.
148 }

```

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

```

149 \msg_new:nnn

```

```

150 { hyp }
151 { empty-info-value }
152 {
153   Empty-value-for-key-#1.\
154   This-isn't-honored-by-all-backends.
155 }

```

## 2 Variants

```

156 \cs_generate_variant:Nn\pdf_destination:nn {nf}
157 \cs_generate_variant:Nn\pdf_object_ref:n {e}
158 \cs_generate_variant:Nn\pdf_pageobject_ref:n {e}

```

## 3 Overwriting/providing commands from hyperref

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

```

159 \chardef\Hy@VersionChecked=1 %don't check the version!
160 %\cs_set_protected:Npn \PDF@SetupDoc{}
161 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

---

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

**\hypercalcbp**

```

162 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

163 \providecommand\@pdfborder{}
164 \providecommand\@pdfborderstyle{}
165 \newcommand\OBJ@OCG@view {} % needed in hyperref
166 \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:

```

167 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
168 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
169 \legacy_if:nT { Hy@setpdfversion }
170 {
171   \msg_warning:nn { hyp }{ pdfversion-disabled }
172 }
173 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

174 \RenewDocumentCommand \Acrobatmenu { m m }
175 {
176   \hyper@linknamed {#1} {#2}
177 }

```

`\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!

```

178 % TODO should go at some time ...
179 % \kv@set@family@handler{Hyp}
180 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
181 \cs_set_protected:Npn \hypersetup #1
182 {
183     %\kvsetkeys{Hyp} {#1}
184     \keys_set:nn { hyp }{ #1 }
185 }
186 % TODO for now unknown keys should only give warnings.
187 \keys_define:nn { hyp }
188 {
189     unknown .code:n =
190     {
191         \msg_warning:nneee { hyp } { unknown-key }
192         { hyp }{ \l_keys_key_str } { #1 }
193     }
194 }

```

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.

```

195 \keys_define:nn { hyp }
196 {
197     next-anchor .code:n =
198     {
199         \AddToHookNext{__hyp/dest/make}
200         {\Hy@MakeCurrentHref{#1}}
201     }
202 }

```

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`)

```

203
204 \bool_new:N \l__hyp_href_url_encode_bool
205 \bool_new:N \l__hyp_href_url_ismap_bool
206 \tl_new:N \l__hyp_href_url_protocol_tl
207 \tl_new:N \l__hyp_href_pdf_destination_tl
208 \tl_new:N \l__hyp_href_pdf_page_tl
209 \tl_new:N \l__hyp_href_run_parameter_tl
210 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
211
212
213 \keys_define:nn { hyp / href }
214 {
215     ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
216     ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
217     ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
218     ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl

```

```

219 ,pdfremotestartview .code:n =
220 {
221   \keys_set:nn { hyp }
222   { pdfremotestartview = #1 }
223 }
224 ,page .code:n =
225 {
226   \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
227   \tl_set:Nn \Hy@href@page {#1}
228 }
229 ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
230 ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
231 ,nextactionraw .code:n =
232 { %perhaps some safety match later, see hyperref code
233   \tl_if_empty:nTF {#1}
234   {
235     \pdfdict_remove:nn{l_hyp/annot/A}{Next}
236   }
237   {
238     \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
239     \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
240     \keys_set:nn {hyp }{ pdfnewwindow = true}
241   }
242 }
243 ,afrelationship .code:n =
244 {
245   \pdfdict_put:nne
246   { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
247 }
248
249 }
250
251 \keys_define:nn { hyp }
252 {
253   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
254   ,href / urlencode .default:n = {true}
255   ,href / urlencode .initial:n = {false}
256   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
257   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
258   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
259 }
260
261 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
262
263 \DeclareRobustCommand*{\href}[1][ ]{%
264   \mode_leave_vertical:
265   \hook_use:n{cmd/href/before}
266   \group_begin:
267   \keys_set:nn { hyp / href } {#1}
268   \bool_if:NTF \l__hyp_href_url_encode_bool
269   {
270     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
271   }
272   {

```

```

273     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
274   }
275   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
276 }
277
278 \begingroup
279   \catcode'\$=6 %
280   \catcode'\#=12 %
281   \gdef\href@\$1{\expandafter\href@split$1##\}%
282   \gdef\href@split$1#$2#$3\\$4{%
283     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
284     \endgroup
285     \hook_use:n{cmd/href/after}
286   }%
287 \endgroup
288
289 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
290
291 \DeclareRobustCommand*{\hrefurl}[1] []
292 {
293   \mode_leave_vertical:
294   \hook_use:n{cmd/href/before}
295   \group_begin:
296   \keys_set:nn { hyp / href } {#1}
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   \hyper@normalise\__hyp_href_pdf_aux:nn
321 }
322
323 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
324 {
325   \exp_args:Nno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
326   \group_end:

```

```

327     \hook_use:n{cmd/hrefpdf/after}
328   }
329
330 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
331 \DeclareRobustCommand*{\hrefrun}[1] []
332 {
333   \mode_leave_vertical:
334   \hook_use:n{cmd/hrefrun/before}
335   \group_begin:
336   \keys_set:nn { hyp / href } {#1}
337   \hyper@normalise\__hyp_href_run_aux:nn
338 }
339
340 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
341 {
342   \exp_args:Nno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
343   \group_end:
344   \hook_use:n{cmd/hrefrun/after}
345 }
346
347
348 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
349
350 \DeclareRobustCommand*{\url}[1] []
351 {
352   \mode_leave_vertical:
353   \hook_use:n{cmd/url/before}
354   \group_begin:
355   \keys_set:nn { hyp / href } {#1}
356   \bool_if:NTF \l__hyp_href_url_encode_bool
357   {
358     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
359   }
360   {
361     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
362   }
363   \hyper@normalise\__hyp_href_url_aux:n
364 }
365
366 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
367 {
368   \exp_args:Nno
369   \hyper@linkurl{\__hyp_href_url_format: {#1}}
370   {\l__hyp_href_url_protocol_tl#1}
371   \group_end:
372   \hook_use:n{cmd/url/after}
373 }
374

```

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

```

375
376 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
377   \def\url@HyperHook##1\endgroup{\url@def{#1}{#2}}%

```

```

378 % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
379 \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
380 #2}
381

```

make the new commands compatible with `\pdfstringdef`:

```

382 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
383 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
384 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
385 \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.

```

386 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
387 {
388   %\tl_set:cn {@#1}{#2}
389   \AddToDocumentProperties[hyperref]{#1}{#2}
390 }
391 \cs_generate_variant:Nn \__hyp_store_metadata:nn {en,ne,ee,no,eo}

```

### 4.2 citecolor

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

```

392 \hook_new:n{hyp/link/cite}
393 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
394 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
395 \keys_define:nn { hyp }
396 {
397   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init}
398   ,citebordercolor
399   .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_init}
400 }
401 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
402 {
403   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
404   {
405     \keys_set:nn { hyp }
406     {
407       linkcolor = hyp/color/cite
408     }
409   }
410   \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
411 }
412 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
413 {
414   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
415   {

```

```

416     \keys_set:nn { hyp }
417     {
418         linkbordercolor      = hyp/color/citeborder
419     }
420 }
421 \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
422 }
423

```

## 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

```

424 \bool_lazy_and:nnF
425 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
426 { \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

```

427 \AddToHook { package/bookmark/after}
428 {
429     \define@key{BKM}{color}
430     {
431         \tl_if_blank:nTF {#1}
432         { \cs_set_eq:NN\BKM@color\@empty }
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 }
442 \legacy_if:nTF { Hy@bookmarks }
443 {
444     \AddToHook{begindocument/before}[hyperref/bookmark]
445     {
446         \RequirePackage{bookmark}
447     }
448 }

```

empty hook chunk to ensure that the chunk exists.

```

449 {
450     \AddToHook{begindocument/before}[hyperref/bookmark]{}
451 }
452 \legacy_if:nT {Hy@draft}
453 {
454     \PassOptionsToPackage{draft}{bookmark}
455 }

```

## 6 Reference and label commands

This uses the in-built property module.

```
\_hyp\_property\_record:nn
```

```
456 %
```

A label command which adds the space commands from LaTeX:

```
457 \cs_new_protected:Npn \_hyp\_property\_record:nn #1 #2 %label/attributes
458 {
459   \@bsphack
460   \property\_record:nn{#1}{#2}
461   \@esphack
462 }
```

we generate a few variants. We use ee-variants as they already exist in the module and once this is there it can go here.

```
463 \cs_generate_variant:Nn \_hyp\_property\_record:nn {ee}
```

*(End of definition for \\_hyp\\_property\\_record:nn.)*

## 7 Variables

### 7.1 Private temporary variables

At first a few generic tmp variables

```
\l\_hyp\_tmpa\_tl
\l\_hyp\_tmpb\_tl 464 \box\_new:N \l\_hyp\_tmpa\_box
\l\_hyp\_tmpa\_seq 465 \tl\_new:N \l\_hyp\_tmpa\_tl
\l\_hyp\_tmpa\_int 466 \tl\_new:N \l\_hyp\_tmpb\_tl
\l\_hyp\_tmpa\_box 467 \seq\_new:N \l\_hyp\_tmpa\_seq
\l\_hyp\_tmpa\_str 468 \int\_new:N \l\_hyp\_tmpa\_int
469 \str\_new:N \l\_hyp\_tmpa\_str
```

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

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

```
\l\_hyp\_dest\_name\_tmpa\_tl  TODO: document and check use!
\l\_hyp\_uri\_tmpa\_tl 470 \tl\_new:N \l\_hyp\_dest\_name\_tmpa\_tl
\l\_hyp\_filename\_tmpa\_tl 471 \tl\_new:N \l\_hyp\_uri\_tmpa\_tl
\_hyp\_text\_tmpa\_str\_uuu\_g\_hyp\_text\_tmpa\_str 472 \tl\_new:N \l\_hyp\_filename\_tmpa\_tl
473 \tl\_new:N \l\_hyp\_para\_tmpa\_tl
474 \str\_new:N \l\_hyp\_text\_tmpa\_str
475 \str\_new:N \g\_hyp\_text\_tmpa\_str
```

*(End of 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.

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

*(End of definition for `\c__hyp_dest_undefined_tl`.)*

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

```
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop
477 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
478 {url,link,file,menu,run}
479 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
480 {
481   URI    = url,
482   GoTo   = link,
483   GoToR  = file,
484   Named  = menu,
485   Launch = run
486 }
487 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
488 {
489   url    = URI,
490   link   = GoTo,
491   file   = GoToR,
492   menu   = Named,
493   run    = Launch
494 }
495
```

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

## 7.3 Variables

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

```
\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_tl
496 \tl_new:N \g__hyp_dest_pdfstartpage_tl
497 \tl_new:N \g__hyp_dest_pdfstartview_tl
498 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

*(End of 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
499 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
500 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
501 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
502 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
503 \tl_new:N \l__hyp_text_enc_file_print_tl
504 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

505
506 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
507 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
508 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
509 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
510 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
511 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

(End of definition for \l\_\_hyp\_text\_enc\_uri\_print\_tl and others.)

It is also unclear how the /Contents entry would look at best. So we use sockets. The first argument is the target (url or destination), For GoTo we also pass the text as argument. The sockets should define \l\_\_hyp\_link\_Contents\_tl

```

512 \tl_new:N\l__hyp_link_Contents_tl
513 \socket_new:nn {hyp/link/GoTo/Contents}{2}
514 \socket_new:nn {hyp/link/URI/Contents}{1}
515 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
516 {
517   \__hyp_text_pdfstring:eoN
518   { Go-to-destination~#1 }
519   { \l__hyp_text_enc_info_print_tl }
520   \l__hyp_link_Contents_tl
521 }
522 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
523 {
524   \__hyp_text_pdfstring:eoN
525   { #1 }
526   { \l__hyp_text_enc_info_print_tl }
527   \l__hyp_link_Contents_tl
528 }
529 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
530 \socket_assign_plug:nn{hyp/link/URI/Contents}{default}

```

\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.

```

531 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End of 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**

```

532 \str_new:N \g__hyp_bordercolormodel_str

```

(End of definition for \g\_\_hyp\_bordercolormodel\_str.)

## 7.4 Booleans

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

```
533 \seq_map_inline:Nn \c__hyp_annot_types_seq
534 {
535   \bool_new:c {l_hyp_annot_color#1_bool}
536 }
```

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

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

```
537 \seq_map_inline:Nn \c__hyp_annot_types_seq
538 {
539   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
540 }
```

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

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

```
541 \seq_map_inline:Nn \c_pdfannot_link_types_seq
542 {
543   \bool_new:c {l__hyp_annot_#1_bool}
544   \bool_set_true:c {l__hyp_annot_#1_bool}
545 }
```

(End of 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.

```
546 \box_new:N \l__hyp_dest_box
```

(End of 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.

```
547 \regex_const:Nn \c__hyp_dest_startview_regex
548 {
549   \A\ *
550   (?
551     (?XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
552   |
553   (?Fit\b|FitB\b)
554   |
555   (?:\bFitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1}
556   |
557   (?FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
558   )
559 }
```

(End of definition for `\c__hyp_dest_startview_regex`.)

## 7.7 PDF dictionaries

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

```
560 \pdfdict_new:n {l__hyp_page/Trans}
561 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of 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`

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

(End of 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`

```
563 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
564 {
565   \str_set:Ne #2 {\text_purify:n { #1 } }
566 }
```

(End of 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`

```
567 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
568 {
569
570 }
```

(End of 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 `l3pdftools`.

<code>#2</code> is str variable, <code>#1</code> should be one of	
<code>utf8/string</code>	<code>(lit) (utf8/string)</code>
<code>utf8/string-raw</code>	<code>lit (utf8/string)</code>
<code>utf8/URI</code>	<code>(percent encoded url)</code>
<code>utf8/URI-raw</code>	<code>percent encoded url</code>
<code>utf16/hex</code>	<code>&lt;HEX&gt; (utf16/hex)</code>
<code>utf16/hex-raw</code>	<code>HEX (utf16/hex)</code>
<code>utf16/string</code>	<code>(lit) (utf16/string)</code>
<code>utf16/string-raw</code>	<code>lit (utf16/string)</code>

`\_hyp\_text\_string\_from\_unicode:nN`

```
571 \cs_new_protected:Npn \_hyp\_text\_string\_from\_unicode:nN #1 #2
572 {
573   \pdf\_string\_from\_unicode:nVN { #1 } #2 #2
574 }
```

*(End of 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`

```
575 \cs_new_protected:Npn \_hyp\_text\_pdfstring:nnN #1 #2 #3
576 {
577   \group\_begin:
578   \Hy@pdfstringtrue
579   \hook\_use:n {hyp/text/pdfstring}
580   \_hyp\_text\_purify:nN { #1 } \l\_hyp\_text\_tmpa\_str
581   \_hyp\_text\_cleanup:N \l\_hyp\_text\_tmpa\_str
582   \_hyp\_text\_string\_from\_unicode:nN { #2 } \l\_hyp\_text\_tmpa\_str
583   \str\_gset\_eq:NN \g\_hyp\_text\_tmpa\_str\l\_hyp\_text\_tmpa\_str
584   \group\_end:
585   \str\_set\_eq:NN #3 \g\_hyp\_text\_tmpa\_str
586 }
587 \cs\_generate\_variant:Nn \_hyp\_text\_pdfstring:nnN {enN,onN,eoN,ooN,noN}
```

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

!!! temporary until all instances are gone

```
588 \cs\_new\_protected:Npn\Hy@pstringdef #1 #2
589 { \_hyp\_text\_pdfstring:enN {#2} {utf8/string-raw}#1 }
```

This is a special version for info keys:

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

```
590 \cs\_new\_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
591 {
592   \_hyp\_text\_pdfstring:noN { #1 }{\l\_hyp\_text\_enc\_info\_print\_tl } #2
593 }
594 \cs\_generate\_variant:Nn \_hyp\_text\_pdfstring\_info:nN {eN,oN}
```

*(End of 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:** `hxtex.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-31 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
595 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
596   {
597   \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums [\HyPL@Labels]>>}
598   }
599
600 \def\Hy@PutCatalog #1 {}
601
602
603 \legacy_if:nT { Hy@pdfpagelabels }
604   {
605     \cs_set_protected:Npn \HyPL@StorePageLabel #1
606       {
607         \tl_gput_right:Nc \HyPL@Labels { \the\Hy@abspage<<#1>> }
608         \__hyp_PageLabels_gpush:
609       }
610   }
```

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

## 10 Core Hyperref Commands

Every `hyperref` has to define eight core command:

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

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

## 10.1 Link level

Links can be nested. Inner links need perhaps special handling, e.g. to deactivate the link, or to change the border, or in the case of tagging to add some additional structure to handle the parent-child rules. We therefore add a global counter which is increased at the begin of link and decreased at the end.

`g__hyp_linknestlevel_int`

```

611 \int_new:N \g__hyp_linknestlevel_int
(End of definition for g__hyp_linknestlevel_int.)
612 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
613 {
614   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
615   { \prg_return_false: }
616   { \prg_return_true: }
617 }
618 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
619 {
620   \use_i:nm {#1}{#2}
621 }
622 \keys_define:nn { hyp }
623 {
624   nested-links .choice:,
625   nested-links / true .code:n =
626     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nm },
627   nested-links / false .code:n =
628     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
629   nested-links .default:n = {true}
630 }

```

## 10.2 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 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`

```

631 \Hy@WrapperDef \__hyp_destination:nn #1 #2
632 {
633   \mode_if_horizontal:T { \@savsf\spacefactor }
634   \Hy@SaveLastskip      %defined in hyperref
635   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging

```

```

636   \_hyp_text_pdfstring:eoN
637     { \HyperDestNameFilter{#1} }
638     { \l__hyp_text_enc_dest_tl }
639     \l__hyp_tmpa_tl
640   \str_if_eq:nnTF {#2} {fitrbox}
641     {
642       \exp_args:NV
643       \pdf_destination:nnnn \l__hyp_tmpa_tl
644       { \box_wd:N \l__hyp_dest_box }
645       { \box_ht:N \l__hyp_dest_box }
646       { \box_dp:N \l__hyp_dest_box }
647     }
648     {
649       \exp_args:NV
650       \pdf_destination:nf
651       { \l__hyp_tmpa_tl }
652       { #2 }
653     }
654   \Hy@RestoreLastskip %defined in hyperref
655   \mode_if_horizontal:T { \spacefactor\@savsf }
656 }

```

(End of 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 657 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 658 \hook_new:n{hyp/anchor}
hyp/anchor 659 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 660 {
661   \exp_args:NnV
662   \_hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
663   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
664   \hook_use:n{hyp/anchor}
665 }
666
667 \cs_new_protected:Npn \hyper@anchorstart #1
668 {
669   \Hy@activeanchortrue
670   \exp_args:NnV
671   \_hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
672   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
673   \hook_use:n{hyp/anchor}
674 }
675
676 \cs_new_protected:Npn \hyper@anchorend
677 {
678   \Hy@activeanchorfalse
679 }

```

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

### 10.3 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:

```

680 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
681   {
682     \mode_leave_vertical:
683     \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
684     \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
685       {
686         \msg_warning:nne
687           { hyp }
688           { empty-destination-name }
689           { \c__hyp_dest_undefined_tl }
690         \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
691       }
692       {
693         \__hyp_text_pdfstring:eoN
694         { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
695         { \l__hyp_text_enc_dest_tl }
696         \l__hyp_dest_name_tmpa_tl
697       }
698     \exp_args:No
699     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
700   }
701
702 \cs_new_protected:Npn \__hyp_link_goto_end:
703   {
704     \pdfannot_link_goto_end:
705   }

```

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

---

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

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

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

```

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

```

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

---

```

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

```

---

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

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

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

```

706
707 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
708 {
709   \bool_if:NTF \l__hyp_annot_GoTo_bool
710   {
711     \int_gincr:N\g__hyp_linknestlevel_int
712     \__hyp_check_link_nesting:TF
713     {
714       \Hy@VerboseLinkStart{#1}{#2}
715       \group_begin:
716       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
717
718       this socket defines \l__hyp_link_Contents_tl
719       \socket_use:nmn{hyp/link/GoTo/Contents}{#2}{#3}
720       \pdfannot_dict_put:nne {link/GoTo}{Contents}
721       {\l__hyp_link_Contents_tl}
722       \hook_use:n {hyp/link/#1}
723       \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
724       \__hyp_link_goto_end:
725       \group_end:
726       \Hy@VerboseLinkStop
727     }
728     {
729       \group_begin: #3\group_end:
730     }
731     \int_gdecr:N\g__hyp_linknestlevel_int
732   }
733   {\let\protect\relax#3}}
734
735 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
736 {
737   \bool_if:NT \l__hyp_annot_GoTo_bool
738   {
739     \int_gincr:N\g__hyp_linknestlevel_int
740     \__hyp_check_link_nesting:TF
741     {
742       \Hy@VerboseLinkStart{#1}{#2}% only for debug
743       \group_begin:
744       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
745       \hook_use:n {hyp/link/#1}
746       \__hyp_link_goto_begin:nw {#2}
747     }
748     {
749       \group_begin:

```

```

748     }
749   }
750 }
751
752 \cs_new_protected:Npn \hyper@linkend
753 {
754   \bool_if:NT \l__hyp_annot_GoTo_bool
755   {
756     \__hyp_check_link_nesting:TF
757     {
758       \__hyp_link_goto_end:
759       \group_end:
760       \Hy@VerboseLinkStop
761     }
762     {
763       \group_end:
764     }
765     \int_gdecr:N\g__hyp_linknestlevel_int
766   }
767 }

```

## 10.4 URI links

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

```

768 \pdfdict_new:n {l_hyp/annot/A/URI}
769 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
770 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
771
772 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
773 {
774   \bool_if:NTF \l__hyp_annot_URI_bool
775   {
776     \int_gincr:N\g__hyp_linknestlevel_int
777     \__hyp_check_link_nesting:TF
778     {
779       \group_begin:
780       \__hyp_text_pdfstring:eoN
781       { #2}
782       { \l__hyp_text_enc_uri_print_tl }
783       \l__hyp_uri_tmpa_tl
784       \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
785       \bool_if:NT \l__hyp_href_url_ismap_bool
786       {
787         \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
788       }

```

This socket defines \l\_\_hyp\_link\_Contents\_tl

```

789     \socket_use:nn{hyp/link/URI/Contents}{#2}
790     \pdfannot_dict_put:nne {link/URI}{Contents}{\l__hyp_link_Contents_tl}
791     \cs_set_eq:NN \# \c_hash_str
792     \cs_set_eq:NN \% \c_percent_str
793     \Hy@safe@activetrue
794     \mode_leave_vertical:
795     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}

```

```

796     \pdfannot_link:nen { URI }
797     {
798     }
799     {
800     \let\protect\relax
801     #1
802     \Hy@xspace@end
803     \Hy@VerboseLinkStop %where is the start??
804     }
805     \group_end:
806     }
807     {
808     \group_begin: #1 \group_end:
809     }
810     \int_gdecr:N\g__hyp_linknestlevel_int
811     }
812     {{\let\protect\relax#1}}
813   }
814

```

## 10.5 GoToR Links files

```

815 \pdfdict_new:n {l_hyp/annot/A/GoToR}
816 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
817 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
818
819 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
820 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
821 {
822   \bool_if:NTF \l__hyp_annot_GoToR_bool
823   {
824     \int_gincr:N\g__hyp_linknestlevel_int
825     \__hyp_check_link_nesting:TF
826     {
827       \group_begin:
828       \tl_set:Ne \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
829       \exp_args:Ne
830       \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
831       {
832         \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n
833         \pdffile_embed_file:noe
834         {}
835         {\l__hyp_filename_tmpa_tl }
836         {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
837       }
838       \pdfdict_put:nne
839       {l_hyp/annot/A/GoToR}
840       {F}
841       {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
842       \__hyp_text_pdfstring:nnN
843       { #3 }
844       { \l__hyp_text_enc_dest_print_tl }
845       \l__hyp_dest_name_tmpa_tl
846       \tl_if_blank:eTF {#3}

```

```

847     {
848         \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
849         {
850             [
851                 \int_eval:n
852                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
853                 /\l__hyp_dest_pdfremotestartview_tl
854             ]
855         }
856     }
857     {
858         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
859     }
860     \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>

```

861     \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
862     \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
863     \pdfannot_link:nnn %expansion??
864     { GoToR }
865     {
866     }
867     {
868         \let\protect\relax
869         #1\Hy@xspace@end
870         \Hy@VerboseLinkStop %where is the start??
871     }
872     \group_end:
873     }
874     {
875         \group_begin: #1 \group_end:
876     }
877     \int_gdecr:N\g__hyp_linknestlevel_int
878     }
879     {{\let\protect\relax#1}}
880     }

```

## 10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

881 \pdfdict_new:n {l_hyp/annot/A/Launch}
882 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
883 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
884
885 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
886 {
887     \bool_if:NTF \l__hyp_annot_Launch_bool
888     {
889         \int_gincr:N\g__hyp_linknestlevel_int
890         \__hyp_check_link_nesting:TF
891         {
892             \group_begin:
893             \__hyp_text_pdfstring:nnN
894             { #1 }

```

```

895         { \l__hyp_text_enc_file_print_tl }
896         \l__hyp_filename_tmpa_tl
897     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
898     \__hyp_text_pdfstring:non
899     { #3 }
900     { \l__hyp_text_enc_para_print_tl }
901     \l__hyp_para_tmpa_tl
902     \bool_if:nTF
903     {
904         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}{}
905         ||
906         \pdf_version_compare_p:Nn > {1.9}
907     }
908     {
909         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
910     }
911     {
912         \pdfdict_put:nne
913         {l_hyp/annot/A/Launch}
914         {Win}
915         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
916     }
917     \mode_leave_vertical:
918     \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
919     \pdfannot_link:nen
920     { Launch }
921     {
922         % /A
923         % <<
924         % \pdfdict_use:n {l_hyp/annot/A/Launch}
925         % >>
926     }
927     {
928         \let\protect\relax
929         #2\Hy@xspace@end
930         \Hy@VerboseLinkStop %where is the start??
931     }
932     \group_end:
933     }
934     { \group_begin: #2 \group_end: }
935     \int_gdecr:N\g__hyp_linknestlevel_int
936     }
937     {{\let\protect\relax#2}}
938     }

```

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

```

939 \use:e
940 { % filename, anchor text, linkname
941     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
942 }
943 {
944     \hyper@linklaunch {#1}{#2}{#3}
945 }

```

## 10.7 Named links (menu)

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

```
946 \pdfdict_new:n {l_hyp/annot/A/Named}
947 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
948 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
949
950 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
951 {
952   \bool_if:NTF \l__hyp_annot_Named_bool
953   {
954     \int_gincr:N\g__hyp_linknestlevel_int
955     \__hyp_check_link_nesting:TF
956     {
957       \group_begin:
958       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
959       {
960         \mode_leave_vertical:
961         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
962           {\pdf_name_from_unicode_e:n{#1}}
963         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
964         \pdfannot_link:nnn { Named }
965         {
966           % /A
967           % <<
968           % \pdfdict_use:n { l_hyp/annot/A/Named }
969           % >>
970         }
971         {
972           #2
973           \Hy@xspace@end
974           \Hy@VerboseLinkStop
975         }
976       }
977       {
978         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
979         #2
980       }
981       \group_end:
982     }
983     { \group_begin: #2 \group_end: }
984     \int_gdecr:N\g__hyp_linknestlevel_int
985   }
986   {\let\protect\relax#2}}
987 }
988
```

## 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.

```
989 \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.
```

```
990 \cs_new_protected:Npn \_hyp_color_select:n #1
991 {
992   \tl_if_head_eq_charcode:nNTF {#1}[ %]
993   {
994     \_hyp_color_select_aux:wn #1
995   }
996   {
997     \color_select:n {#1}
998   }
999 }
1000
1001 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
1002 {
1003   \color_select:nn {#1}{#2}
1004 }
1005
1006 \cs_generate_variant:Nn \_hyp_color_select:n {e}
```

*(End of 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.
```

```
1007 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
1008 {
1009   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1010   {
1011     \_hyp_color_set_aux:nwn { #1 } #2
1012   }
1013   {
1014     \color_set:nn {#1} {#2}
1015   }
1016 }
1017
1018 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
1019 {
1020   \color_set:nnn {#1}{#2}{#3}
1021 }
1022
1023 \cs_generate_variant:Nn \_hyp_color_set:nn {ne}
```

(End of 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.

```
1024 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1025   {
1026     \hook_gput_code:nnn
1027       {pdfannot/link/#2/begin}
1028       {hyp/color}
1029     {
1030       \bool_if:cT { l_hyp_annot_color#1_bool }
1031       {
1032         \group_begin:
1033         \color_select:n { hyp/color/#1}
1034       }
1035     }
1036     \hook_gput_code:nnn
1037       {pdfannot/link/#2/end}
1038       {hyp/color}
1039     {
1040       \bool_if:cT { l_hyp_annot_color#1_bool }
1041       {
1042         \group_end:
1043       }
1044     }
1045   }
```

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

```
1046 \keys_define:nn { hyp }
1047   {
1048     ,colorlinks .choice:
1049     ,colorlinks / true .meta:n =
1050     {
1051       ,pdfborder={0~0~0}
1052       ,pdfborderstyle=
1053       ,colorurl =#1
1054       ,colorlink =#1
1055       ,colorryn =#1
1056       ,colormenu =#1
1057       ,colorfile =#1
1058     }
1059     ,colorlinks / false .meta:n =
1060     {
1061       ,colorurl =#1
1062       ,colorlink =#1
1063       ,colorryn =#1
1064       ,colormenu =#1
1065       ,colorfile =#1
1066     }
1067     ,colorlinks .default:n = {true}
1068   }
```

```

colorurl (setup key)
colorlink (setup key) 1069 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 1070 {
colormenu (setup key) 1071   \keys_define:nn { hyp }
colorfile (setup key) 1072   {
  urlcolor (setup key) 1073     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1074     ,#1color .code:n =   { \__hyp_color_set:ne {hyp/color/#1}{##1} }
  runcolor (setup key) 1075   }
menucolor (setup key) 1076 }
filecolor (setup key) 1077 \keys_define:nn { hyp }
allcolors (setup key) 1078 {
 1079   ,allcolors .meta:n =
 1080   {
 1081     ,urlcolor=#1
 1082     ,linkcolor=#1
 1083     ,runcolor=#1
 1084     ,filecolor=#1
 1085     ,menucolor=#1
 1086   }
 1087   ,allcolors .value_required:n = true
 1088 }
1089 }

```

## 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)

```

1090 \keys_define:nn { hyp }
1091 {
1092   ,bordercolormodel .choices:nn =
1093   {rgb,cmyk}
1094   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1095   ,bordercolormodel .initial:n ={rgb}
1096 }

1097 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1098 {
1099   \keys_define:nn { hyp }
1100   {
1101     #1bordercolor .code:n =
1102     {
1103       \tl_if_empty:nTF { ##1 }
1104       {
1105         \pdfannot_dict_remove:nn
1106         {link/#2}
1107         { C }
1108       }
1109       {
1110         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1111         \color_export:nVN

```

```

1112         {hyp/color/#1border}
1113         \g__hyp_bordercolormodel_str
1114         \l__hyp_tmpa_tl
1115         \pdfannot_dict_put:nne
1116         {link/#2}
1117         { C }
1118         { [\l__hyp_tmpa_tl] }
1119     }
1120 }
1121 }
1122 }
1123
1124 \keys_define:nn { hyp }
1125 {
1126     ,allbordercolors .meta:n =
1127     {
1128         ,linkbordercolor=#1
1129         ,urlbordercolor =#1
1130         ,filebordercolor=#1
1131         ,menubordercolor=#1
1132         ,runbordercolor =#1
1133     }
1134     ,allbordercolors .value_required:n = true
1135 }
1136

```

### 11.3.2 Borderwidth and -arc

```

1137 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1138 {
1139     \keys_define:nn { hyp }
1140     {
1141         #1border .code:n =
1142         {
1143             \tl_if_empty:nTF { ##1 }
1144             {
1145                 \pdfannot_dict_remove:nn
1146                 {link/#2}
1147                 { Border }
1148             }
1149             {
1150                 \pdfannot_dict_put:nnn
1151                 {link/#2}
1152                 { Border }
1153                 { [##1] }
1154             }
1155         }
1156     }
1157 }
1158 \keys_define:nn { hyp }
1159 {
1160     ,pdfborder .code:n =
1161     {
1162         \tl_if_empty:nTF { #1 }

```

```

1163     {
1164         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1165         {
1166             \pdfannot_dict_remove:nn
1167             {link/##2}
1168             { Border }
1169         }
1170     }
1171     {
1172         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1173         {
1174             \pdfannot_dict_put:nnn
1175             {link/##2}
1176             { Border }
1177             { [#1] }
1178         }
1179     }
1180 }
1181 ,pdfborder .initial:n = {0~0~1},
1182 }

```

### 11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1183 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1184 {
runborderstyle (setup key) 1185   \keys_define:nn { hyp }
fileborderstyle (setup key) 1186   {
menuborderstyle (setup key) 1187     #1borderstyle .code:n =
1188     {
1189         \tl_if_empty:nTF { ##1 }
1190         {
1191             \pdfannot_dict_remove:nn
1192             {link/#2}
1193             { BS }
1194         }
1195         {
1196             \pdfannot_dict_put:nnn
1197             {link/#2}
1198             { BS }
1199             { <<##1>> }
1200         }
1201     }
1202 }
1203 }
1204 \keys_define:nn { hyp }
1205 {
1206     ,pdfborderstyle .code:n =
1207     {
1208         \tl_if_empty:nTF { #1 }
1209         {
1210             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1211             {

```

```

1212         \pdfannot_dict_remove:nn
1213         {link/##2}
1214         { BS }
1215     }
1216 }
1217 {
1218     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1219     {
1220         \pdfannot_dict_put:nnn
1221         {link/##2}
1222         { BS }
1223         { <<#1>> }
1224     }
1225 }
1226 }
1227 ,pdfborderstyle .initial:n = {},
1228 }

```

## 11.4 ocgcolorlinks

OCG colorlinks need objects and an entry in the catalog. Perhaps the objects need public names to avoid that ocg2 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.

```

1229 \cs_new_protected:Npn \__hyp_ocg_init:
1230 {
1231     \pdf_object_new:n { __hyp/OCG/View }
1232     \pdf_object_new:n { __hyp/OCG/Print }
1233     \pdf_object_new:n { __hyp/OCG/config }
1234     \pdf_object_new:n { __hyp/OCG/refarray }
1235     \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1236     {
1237         \pdf_object_ref:n { __hyp/OCG/View }
1238         \c_space_tl
1239         \pdf_object_ref:n { __hyp/OCG/Print }
1240     }
1241     \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1242     {
1243         /Type/OCG
1244         /Name(View)
1245         /Usage
1246         <<
1247         /Print <</PrintState/OFF>>~
1248         /View <</ViewState/ON >>~
1249         >>
1250     }
1251     \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1252     {
1253         /Type/OCG
1254         /Name(Print)
1255         /Usage
1256         <<
1257         /Print <</PrintState/ON>>~

```

```

1258         /View <</ViewState/OFF>>~
1259     >>
1260     }
1261     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG} }
1262     \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG} }
1263     \pdf_object_write:nne { __hyp/OCG/config } { dict }
1264     {
1265         /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
1266         /AS[
1267             <<
1268                 /Event/View
1269                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1270                 /Category[/View]
1271             >>
1272             <<
1273                 /Event/Print
1274                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1275                 /Category[/Print]
1276             >>
1277             <<
1278                 /Event/Export
1279                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1280                 /Category[/Print]
1281             >>
1282             ]
1283         }
1284     \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG} }
1285     \cs_gset:Npn \__hyp_ocg_init: {}
1286 }

```

(End of 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.

```

1287 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1288 {
1289     \hook_gput_code:nnn
1290     {pdfannot/link/#2/begin}
1291     {hyp/ocg}
1292     {
1293         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1294         {
1295             \__hyp_ocg_init:
1296             \group_begin:
1297             \hbox_set:Nw \l__hyp_tmpa_box
1298         }
1299     }
1300     \hook_gput_code:nnn
1301     {pdfannot/link/#2/end}
1302     {hyp/ocg}
1303     {
1304         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1305         {
1306             \hbox_set_end:
1307             \mbox

```

```

1308         {
1309             \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1310             \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1311             \pdf_emc:
1312             \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1313             \group_begin:
1314             \color_select:n { hyp/color/#1 }
1315             \box_use_drop:N \l__hyp_tmpa_box
1316             \group_end:
1317             \pdf_emc:
1318         }
1319     \group_end:
1320 }
1321 }
1322 }

```

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

```

ocgcolorlink (setup key) 1323 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1324 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1325 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1326 {
ocgcolorrund (setup key) 1327     \keys_define:nn { hyp }
1328     {
1329         ,_ocgcolorlinks .meta:n =
1330         {
1331             ocgcolorlink=#1,
1332             ocgcolorurl=#1,
1333             ocgcolorfile=#1,
1334             ocgcolorrund=#1,
1335             ocgcolormenu=#1
1336         }
1337         ,_ocgcolorlinks .default:n = true
1338     }
1339 }
1340 {
1341     \keys_define:nn { hyp }
1342     {
1343         ,_ocgcolorlinks .code:n =
1344         {
1345             \msg_warning:nnee
1346             { hyp }
1347             { ignore-deprecated-or-unknown-option-in-pdf-version }
1348             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1349         }
1350     }
1351 }
1352
1353 \keys_define:nn { hyp }
1354 {
1355     ,ocgcolorlinks .choice:
1356     ,ocgcolorlinks / true .meta:n =
1357     {
1358         pdfborder      ={{0~0~0}},
1359         pdfborderstyle ={ },

```

```

1360     colorlinks      = false,
1361     _ocgcolorlinks = true
1362   }
1363   ,ocgcolorlinks / false .meta:n =
1364   {
1365     _ocgcolorlinks = false
1366   }
1367   ,ocgcolorlinks .default:n = {true}
1368 }
1369
1370 \seq_map_inline:Nn \c__hyp_annot_types_seq
1371 {
1372   \bool_lazy_or:nnTF
1373   { \pdf_version_compare_p:Nn > {1.4} }
1374   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1375   {
1376     \keys_define:nn { hyp }
1377     {
1378       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1379     }
1380   }
1381   {
1382     \keys_define:nn { hyp }
1383     {
1384       ,ocgcolor#1 .code:n=
1385       {
1386         \msg_warning:nnee
1387         { hyp }
1388         { ignore-deprecated-or-unknown-option-in-pdf-version }
1389         { ocgcolor#1 }
1390         { \pdf_version_major:.\pdf_version_minor: }
1391       }
1392     }
1393   }
1394 }

```

## 11.5 Highlighting

This keys set what happens if you click on a link

```

1395 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1396 {
1397   \keys_define:nn { hyp }
1398   {
1399     ,#1highlight .choices:nn =
1400     { /I, /N, /O, /P}
1401     {
1402       \pdfannot_dict_put:nnn
1403       {link/#2}
1404       { H }
1405       { ##1 }
1406     }
1407     ,#1highlight / .code:n =
1408     {

```

```

1410         \pdfannot_dict_remove:nn
1411             {link/#2}
1412             { H }
1413
1414     }
1415     ,#1highlight / unknown .code:n =
1416     {
1417         \msg_warning:nnee { hyp } { unknown-choice+empty }
1418         { #1highlight }
1419         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1420         { \exp_not:n {##1} }
1421     }
1422 }
1423 }
1424
1425
1426 \keys_define:nn { hyp }
1427 {
1428     ,pdfhighlight .choices:nn =
1429     { /I, /N, /O, /P}
1430     {
1431         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1432         {
1433             \pdfannot_dict_put:nnn
1434             {link/###2}
1435             { H }
1436             { #1 }
1437         }
1438     }
1439     ,pdfhighlight / .code:n =
1440     {
1441         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1442         {
1443             \pdfannot_dict_remove:nn
1444             {link/##2}
1445             { H }
1446         }
1447     }
1448     ,pdfhighlight .initial:n = {/I},
1449     ,pdfhighlight / unknown .code:n =
1450     {
1451         \msg_warning:nnee { hyp } { unknown-choice+empty }
1452         { pdfhighlight }
1453         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1454         { \exp_not:n {##1} }
1455     }
1456 }

```

## 11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1457 \keys_define:nn { hyp }
hideurl (setup key)
hidefile (setup key)
hiderun (setup key)
hidemenu (setup key)

```

```

1458 {
1459   hidelinks .meta:n =
1460   {
1461     ,colorlinks      = false
1462     ,ocgcolorlinks  = false
1463     ,pdfborder       = { 0~0~0 }
1464     ,pdfborderstyle=
1465   }
1466 }
1467
1468 \seq_map_inline:Nn \c__hyp_annot_types_seq
1469 {
1470   \keys_define:nn { hyp }
1471   {
1472     hide#1 .meta:n =
1473     {
1474       ,color#1      = false
1475       ,ocgcolor#1  = false
1476       ,#1border     = { 0~0~0 }
1477       ,#1borderstyle =
1478     }
1479   }
1480 }

```

## 11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```

1481 \keys_define:nn { hyp }
1482 {
1483   colorscheme .code:n =
1484   {
1485     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1486     {
1487       \keys_set:nn { hyp }
1488       {
1489         ##1 = ##2
1490       }
1491     }
1492   }
1493 }
1494 \keys_set:nn { hyp } {colorscheme=phelype}

```

## 12 Keys

### 12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode` (*setup key*)

`pdfencoding` (*setup key*)

`pdfversion` (*setup key*)

```

1495 \keys_define:nn { hyp }
1496 {

```

```

1497     ,unicode      .code:n = {}
1498     ,pdfencoding .code:n = {}
1499     ,pdfversion  .code:n =
1500     {
1501         \msg_warning:nn { hyp }{ pdfversion-disabled }
1502     }
1503 }
1504 %

```

## 12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1505 \keys_define:nn { hyp }
draft (setup key) 1506 {
final (setup key) 1507     ,verbose .legacy_if_set:n = {Hy@verbose}
1508     ,debug .legacy_if_set:n = {Hy@verbose}
1509 }
1510 \keys_define:nn { hyp }
1511 {
1512     ,draft .code:n =
1513     {
1514         \Hy@drafttrue
1515         \PassOptionsToPackage{draft}{bookmark}
1516     }
1517     ,final .code:n =
1518     {
1519         \Hy@finaltrue
1520         \PassOptionsToPackage{final}{bookmark}
1521     }
1522 }

extension (setup key)
hypertextnames (setup key) 1523 \keys_define:nn { hyp }
naturalnames (setup key) 1524 {
pageanchor (setup key) 1525     ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1526     ,extension .initial:n= pdf
linktocpage (setup key) 1527     ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1528     ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1529     ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1530     ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1531     ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1532     ,plainpages .legacy_if_set:n = {Hy@plainpages}
1533 }
1534
1535 \keys_define:nn { hyp }
1536 {
1537     ,linktoc .choices:nn = { none, section, all, page }
1538     {
1539         \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1540     }
1541     ,linktoc / unknown .code:n =
1542     {

```

```

1543     \msg_warning:nneee { hyp } { unknown-choice }
1544     { linktoc }
1545     { none, section, all, page }
1546     { \exp_not:n {#1} }
1547   }
1548   ,linktocpage .choice:
1549   ,linktocpage / true .meta:n = {linktoc=page}
1550   ,linktocpage / false .meta:n = {linktoc=section}
1551   ,linktocpage .default:n = true
1552 }
1553

```

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

```

url (setup key) 1554 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1555 {
menu (setup key) 1556   \keys_define:nn { hyp }
run (setup key) 1557   {
1558     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1559   }
1560 }

```

```

1561 \keys_define:nn { hyp }
1562 {
1563   ,baseurl .code:n =
1564   {
1565     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1566     \tl_if_empty:NTF \l__hyp_tmpa_tl
1567     {
1568       \pdfmanagement_remove:nn {Catalog} { URI }
1569     }
1570     {
1571       \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1572     }
1573     \__hyp_store_metadata:nn {baseurl}{#1}
1574   }
1575   %only false does something ...
1576   ,bookmarks .choice:
1577   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1578   ,bookmarks / true .code:n = {}
1579   ,bookmarks .default:n = {true}
1580   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1581   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1582   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1583   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1584   ,pdfcenterwindow .choice:
1585   ,pdfcenterwindow / false .code:n =
1586   {
1587     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1588   }
1589   ,pdfcenterwindow / true .code:n =
1590   {
1591     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1592   }
1593   ,pdfcenterwindow / .code:n =

```

```

1594     {
1595     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1596     }
1597 ,pdfcenterwindow / unknown .code:n =
1598     {
1599     \msg_warning:nnee { hyp } { no-bool }
1600     { pdfcenterwindow }
1601     { \exp_not:n {#1} }
1602     }
1603 ,pdfcenterwindow .default:n = true
1604 ,pdfdirection .choice:
1605 ,pdfdirection / L2R .code:n =
1606     {
1607     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1608     }
1609 ,pdfdirection / R2L .code:n =
1610     {
1611     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1612     }
1613 ,pdfdirection / .code:n =
1614     {
1615     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1616     }
1617 ,pdfdirection / unknown .code:n =
1618     {
1619     \msg_warning:nneee { hyp } { unknown-choice+empty }
1620     { pdfdirection }
1621     { L2R , R2L }
1622     { \exp_not:n {#1} }
1623     }
1624 ,pdfdisplaydoctitle .choice:
1625 ,pdfdisplaydoctitle / false .code:n =
1626     {
1627     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1628     }
1629 ,pdfdisplaydoctitle / true .code:n =
1630     {
1631     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1632     }
1633 ,pdfdisplaydoctitle .default:n = true
1634 ,pdfduplex .choices:nn =
1635 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1636     {
1637     \pdf_version_compare:NnTF > {1.6}
1638     {
1639     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1640     { PrintDuplex } { /#1 }
1641     }
1642     {
1643     \msg_warning:nnee
1644     {hyp}
1645     {ignore-deprecated-or-unknown-option-in-pdf-version}
1646     {pdfduplex}
1647     {\pdf_version:}

```

```

1648     }
1649   }%
1650 ,pdfduplex / .code:n =
1651   {
1652     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1653   }
1654 ,pdfduplex / unknown .code:n =
1655   {
1656     \msg_warning:nnee { hyp } { unknown-choice+empty }
1657     { pdfduplex }
1658     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1659     { \exp_not:n {#1} }
1660   }
1661 ,pdffitwindow .choice:
1662 ,pdffitwindow / false .code:n =
1663   {
1664     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1665   }
1666 ,pdffitwindow / true .code:n =
1667   {
1668     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1669   }
1670 ,pdffitwindow / .code:n =
1671   {
1672     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1673   }
1674 ,pdffitwindow .default:n = true
1675 ,pdffitwindow / unknown .code:n =
1676   {
1677     \msg_warning:nnee { hyp } { no-bool }
1678     { pdffitwindow }
1679     { \exp_not:n {#1} }
1680   }
1681 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1682 ,pdflinkmargin .initial:n = {1pt}
1683 ,pdfmenubar .choice:
1684 ,pdfmenubar / true .code:n =
1685   {
1686     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1687   }
1688 ,pdfmenubar / false .code:n =
1689   {
1690     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1691     { HideMenubar } { true }
1692   }
1693 ,pdfmenubar / .code:n =
1694   {
1695     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1696   }
1697 ,pdfmenubar .default:n = true
1698 ,pdfmenubar / unknown .code:n =
1699   {
1700     \msg_warning:nnee { hyp } { no-bool }
1701     { pdfmenubar }

```

```

1702         { \exp_not:n {#1} }
1703     }
1704 ,pdfnewwindow .choice:
1705 ,pdfnewwindow / true .code:n =
1706     {
1707         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1708         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1709     }
1710 ,pdfnewwindow / false .code:n =
1711     {
1712         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1713         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1714     }
1715 ,pdfnewwindow / .code:n =
1716     {
1717         \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1718         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1719     }
1720 ,pdfnonfullscreenpagemode .choices:nn =
1721 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1722 {
1723     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1724     { NonFullScreenPageMode } {/#1}
1725 }
1726 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1727 {
1728     \pdf_version_compare:NnTF < {1.6}
1729     {
1730         %message
1731     }
1732     {
1733         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1734         {NonFullScreenPageMode}{/UseAttachments}
1735     }
1736 }
1737 ,pdfnonfullscreenpagemode / .code:n =
1738 {
1739     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1740 }
1741 ,pdfnonfullscreenpagemode / unknown .code:n =
1742 {
1743     \msg_warning:nnee { hyp } { unknown-choice+empty }
1744     { pdfnonfullscreenpagemode }
1745     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1746     { \exp_not:n {#1} }
1747 }
1748 ,pdfnumcopies .code:n =
1749 {
1750     \pdf_version_compare:NnTF > {1.6}
1751     {
1752         \tl_if_empty:nTF {#1}
1753         {
1754             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1755         }

```

```

1756         {
1757             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1758                 {NumCopies}{#1}
1759         }
1760     }
1761     {
1762         \msg_warning:nnee
1763             {hyp}
1764             {ignore-deprecated-or-unknown-option-in-pdf-version}
1765             {pdfnumcopies}
1766             {\pdf_version:}
1767     }
1768 }
1769 ,pdfpagelayout .choices:nn =
1770 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1771 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1772 ,pdfpagelayout / .code:n =
1773 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1774 ,pdfpagelayout / unknown .code:n =
1775 {
1776     \msg_warning:nneee { hyp } { unknown-choice+empty }
1777     { pdfpagelayout }
1778     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1779     { \exp_not:n {#1} }
1780 }
1781 ,pdfpagemode .choices:nn =
1782 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1783 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1784 ,pdfpagemode / UseAttachments .code:n =
1785 {
1786     \pdf_version_compare:NnTF > {1.5}
1787     {
1788         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1789     }
1790     {
1791         \msg_warning:nnee
1792             {hyp}
1793             {ignore-deprecated-or-unknown-value-in-pdf-version}
1794             {UseAttachments}
1795             {\pdf_version:}
1796     }
1797 }
1798 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1799 ,pdfpagemode / unknown .code:n =
1800 {
1801     \msg_warning:nneee { hyp } { unknown-choice+empty }
1802     { pdfpagemode }
1803     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1804     { \exp_not:n {#1} }
1805 }
1806 ,pdfpagescrop .code:n =
1807 {
1808     \tl_if_empty:nTF {#1} %or blank?
1809     {

```

```

1810         \pdfmanagement_remove:nn {Pages} { CropBox }
1811     }
1812     {
1813         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1814     }
1815 }
1816 ,pdfpicktraybypdfsize .choice:
1817 ,pdfpicktraybypdfsize / true .code:n =
1818 {
1819     \pdf_version_compare:NnTF > {1.6}
1820     {
1821         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1822         { PickTrayByPDFSize } { true }
1823     }
1824     {
1825         \msg_warning:nnee
1826         {hyp}
1827         {ignore-deprecated-or-unknown-option-in-pdf-version}
1828         {pdfpicktraybypdfsize}
1829         {\pdf_version:}
1830     }
1831 }
1832 ,pdfpicktraybypdfsize / false .code:n =
1833 {
1834     \pdf_version_compare:NnTF > {1.6}
1835     {
1836         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1837         { PickTrayByPDFSize } { false }
1838     }
1839     {
1840         \msg_warning:nnee
1841         {hyp}
1842         {ignore-deprecated-or-unknown-option-in-pdf-version}
1843         {pdfpicktraybypdfsize}
1844         {\pdf_version:}
1845     }
1846 }
1847 ,pdfpicktraybypdfsize / .code:n =
1848 {
1849     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1850 }
1851 ,pdfpicktraybypdfsize / unknown .code:n =
1852 {
1853     \msg_warning:nnee { hyp } { no-bool }
1854     { picktraybypdfsize }
1855     { \exp_not:n {#1} }
1856 }
1857 ,pdfprintarea .choices:nn =
1858 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1859 {
1860     \pdf_version_compare:NnTF < {2.0}
1861     {
1862         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1863         { PrintArea } { /#1 }

```

```

1864     }
1865     {
1866         \msg_warning:nnee
1867         {hyp}
1868         {ignore-deprecated-or-unknown-option-in-pdf-version}
1869         {pdfprintarea}
1870         {\pdf_version:}
1871     }
1872 }%
1873 ,pdfprintarea / .code:n =
1874 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1875 ,pdfprintarea / unknown .code:n =
1876 {
1877     \msg_warning:nnee { hyp } { unknown-choice+empty }
1878     { pdfprintarea }
1879     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1880     { \exp_not:n {#1} }
1881 }
1882 ,pdfprintclip .choices:nn =
1883 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1884 {
1885     \pdf_version_compare:NnTF < {2.0}
1886     {
1887         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1888         { PrintClip } { /#1 }
1889     }
1890     {
1891         \msg_warning:nnee
1892         {hyp}
1893         {ignore-deprecated-or-unknown-option-in-pdf-version}
1894         {pdfprintclip}
1895         {\pdf_version:}
1896     }
1897 }%
1898 ,pdfprintclip / .code:n =
1899 {
1900     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1901 }
1902 ,pdfprintclip / unknown .code:n =
1903 {
1904     \msg_warning:nnee
1905     { hyp }
1906     { unknown-choice+empty }
1907     { pdfprintclip }
1908     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1909     { \exp_not:n {#1} }
1910 }
1911 ,pdfprintpagerange .code:n =
1912 {
1913     \pdf_version_compare:NnTF > {1.6}
1914     {
1915         \tl_if_empty:nTF { #1}
1916         {
1917             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }

```

```

1918         { PrintPageRange }
1919     }
1920     {
1921         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1922         {PrintPageRange}{[#1]}
1923     }
1924 }
1925 {
1926     \msg_warning:nnee
1927     {hyp}
1928     {ignore-deprecated-or-unknown-option-in-pdf-version}
1929     {pdfprintpagerange}
1930     {\pdf_version:}
1931 }
1932 }
1933 ,pdfprintscaling .choices:nn =
1934 { None, AppDefault }
1935 {
1936     \pdf_version_compare:NnTF > {1.5}
1937     {
1938         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1939         { PrintScaling } { /#1 }
1940     }
1941     {
1942         \msg_warning:nnee
1943         {hyp}
1944         {ignore-deprecated-or-unknown-option-in-pdf-version}
1945         {pdfprintscaling}
1946         {\pdf_version:}
1947     }
1948 }%
1949 ,pdfprintscaling / .code:n =
1950 {
1951     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1952 }
1953 ,pdfprintscaling / unknown .code:n =
1954 {
1955     \msg_warning:nneee { hyp } { unknown-choice+empty }
1956     { pdfprintarea }
1957     { None, AppDefault }
1958     { \exp_not:n {#1} }
1959 }
1960 ,pdfremotestartview .code:n =
1961 {
1962     \tl_set:Ne \l__hyp_tmpa_tl {#1-null-null-null~}
1963     \exp_args:NNV
1964     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1965     {
1966         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1967     }
1968     {
1969         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1970         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1971     }

```

```

1972     }
1973 ,pdfremotestartview .initial:n = {Fit}
1974 % pdfstartpage is special as it shares code with pdfstartview
1975 ,pdfstartpage .code:n =
1976 {
1977     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1978     \bool_if:nTF
1979     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1980     {
1981         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1982     }
1983     {
1984         \pdfmanagement_add:nne {Catalog} { OpenAction }
1985         {
1986             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1987         }
1988     }
1989 }
1990 ,pdfstartpage .initial:n =1
1991 ,pdfstartview .code:n =
1992 {
1993     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1994     \exp_args:NNV
1995     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1996     {
1997         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1998     }
1999     {
2000         \msg_warning:nmmm {hyp}{invalid-destination-value}{#1}{pdfstartview}
2001         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2002     }
2003     \bool_if:nTF
2004     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2005     {
2006         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2007     }
2008     {
2009         \pdfmanagement_add:nne {Catalog} { OpenAction }
2010         {
2011             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2012         }
2013     }
2014 }
2015 ,pdfstartview .initial:n = Fit
2016 ,pdftoolbar .choice:
2017 ,pdftoolbar / true .code:n =
2018 {
2019     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2020 }
2021 ,pdftoolbar / false .code:n =
2022 {
2023     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2024     { HideToolbar } { true }
2025 }

```

```

2026 ,pdftoolbar / true .code:n =
2027 {
2028     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2029 }
2030 ,pdftoolbar .default:n = true
2031 ,pdftoolbar / unknown .code:n =
2032 {
2033     \msg_warning:nnee { hyp } { no-bool }
2034     { pdftoolbar }
2035     { \exp_not:n {#1} }
2036 }
2037 % pdfview see below.
2038 ,pdfviewarea .choices:nn =
2039 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2040 {
2041     \pdf_version_compare:NnTF < {2.0}
2042     {
2043         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2044         { ViewArea } { /#1 }
2045     }
2046     {
2047         \msg_warning:nnee
2048         {hyp}
2049         {ignore-deprecated-or-unknown-option-in-pdf-version}
2050         {pdfviewarea}
2051         {\pdf_version:}
2052     }
2053 }%
2054 ,pdfviewarea / .code:n =
2055 {
2056     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2057 }
2058 ,pdfviewarea / unknown .code:n =
2059 {
2060     \msg_warning:nneee { hyp } { unknown-choice+empty }
2061     { pdfviewarea }
2062     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2063     { \exp_not:n {#1} }
2064 }
2065 ,pdfviewclip .choices:nn =
2066 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2067 {
2068     \pdf_version_compare:NnTF < {2.0}
2069     {
2070         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2071         { ViewClip } { /#1 }
2072     }
2073     {
2074         \msg_warning:nnee
2075         {hyp}
2076         {ignore-deprecated-or-unknown-option-in-pdf-version}
2077         {pdfviewclip}
2078         {\pdf_version:}
2079     }

```

```

2080     }%
2081 ,pdfviewclip / .code:n =
2082     {
2083     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2084     }
2085 ,pdfviewclip / unknown .code:n =
2086     {
2087     \msg_warning:nneee { hyp } { unknown-choice+empty }
2088     { pdfviewclip }
2089     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2090     { \exp_not:n {#1} }
2091     }
2092 ,pdfwindowui .choice:
2093 ,pdfwindowui / true .code:n =
2094     {
2095     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2096     }
2097 ,pdfwindowui / false .code:n =
2098     {
2099     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2100     { HideWindowUI } { true }
2101     }
2102 ,pdfwindowui / .code:n =
2103     {
2104     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2105     }
2106 ,pdfwindowui / unknown .code:n =
2107     {
2108     \msg_warning:nnee { hyp } { no-bool }
2109     { pdfwindowui }
2110     { \exp_not:n {#1} }
2111     }
2112 ,pdfwindowui .default:n = true
2113 }

```

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

```

2114 \keys_define:nn { hyp }
2115 {
2116 ,pdfview .code:n =
2117     {
2118     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2119     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2120     {
2121     { xyz }
2122     {
2123     \int_compare:nNnTF { \seq_count:N \l__hyp_tmpa_seq } > { 1 }
2124     {
2125     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2126     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2127     {
2128     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2129     }
2130     {
2131     \tl_set:Ne \l__hyp_dest_pdfview_tl

```

```

2132         {
2133             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2134         }
2135     }
2136 }
2137 {
2138     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2139 }
2140 }
2141 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2142 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2143 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2144 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2145 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2146 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2147 { fitr }
2148 {
2149     \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2150     {
2151         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2152     }
2153     {
2154         %ensure 4 values ...
2155         \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2156         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2157         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2158         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2159         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2160         {
2161             \fp_eval:n
2162             {
2163                 round
2164                 (
2165                     abs
2166                     (
2167                         \seq_item:Nn\l__hyp_tmpa_seq{4}
2168                         -
2169                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2170                     ),
2171                     3
2172                 )
2173             }bp
2174         }{}
2175         \box_set_dp:Nn \l__hyp_dest_box
2176         {
2177             \fp_eval:n
2178             {
2179                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2180             }bp
2181         }
2182         \box_set_ht:Nn \l__hyp_dest_box
2183         {
2184             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2185         }

```

```

2186         }
2187     }
2188 }
2189 {
2190     \msg_warning:nmn {hyp}{invalid-destination-value}{#1}{pdfview}
2191     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2192 }
2193 }
2194 ,pdfview .initial:n = {xyz}
2195 }

```

## 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.

```

2196 \keys_define:nn { hyp }
2197 {
2198     ,pdflang .code:n =
2199     {
2200         \tl_if_empty:nF { #1 }
2201         {
2202             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2203             \AddToDocumentProperties[document]{lang}{#1}
2204         }
2205     }
2206 }

```

### 12.3.1 “info keys”

**pdfauthor** (*setup key*) The keys store their value also in the metadata container, so that hyperxmp can use them.  
**pdftitle** (*setup key*) Creator and Producer can't be removed with the pdfmanagement, but we allow to set an empty value. If the value begin with an optional argument, we assume a multilanguage  
**pdfcreator** (*setup key*) empty value. If the value begin with an optional argument, we assume a multilanguage  
**pdfsubject** (*setup key*) clist and use only the first value. The values are expanded with `\text_expand:n`  
**pdfproducer** (*setup key*)  
**pdfkeywords** (*setup key*)

```

2207 \regex_new:N\l__hyp_optlang_regex
2208 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\+]\](.*)}
2209 \cs_generate_variant:Nn\clist_item:nn{on}
2210 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2211 {
2212     \keys_define:nn { hyp }
2213     {
2214         pdf#1 .code:n =
2215         {
2216             \tl_set:Ne\l__hyp_tmpa_tl {\text_expand:n{##1}}
2217             \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2218             \tl_if_empty:NTF \l__hyp_tmpa_tl
2219             {
2220                 \str_case:nnF { #1 }
2221                 {
2222                     {creator}
2223                     {
2224                         \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }

```

```

2225         \pdfmanagement_add:nne {Info}{Creator}{()}
2226     }
2227     {producer}
2228     {
2229         \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2230         \pdfmanagement_add:nne {Info}{Producer}{()}
2231     }
2232 }
2233 {
2234     \pdfmanagement_remove:nn {Info}{#2}
2235 }
2236 }
2237 {
2238     \tl_set:Nel__hyp_tmpb_tl {\clist_item:on{\__hyp_tmpa_tl}{1}}
2239     \exp_args:NNV
2240     \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_s
2241     \seq_if_empty:NTF\l__hyp_tmpa_seq
2242     {
2243         \__hyp_text_pdfstring_info:oN {\__hyp_tmpa_tl}\l__hyp_tmpa_str
2244     }
2245     {
2246         \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2247     }
2248     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2249     {
2250         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2251     }
2252 }
2253 }
2254 }
2255 \keys_define:nn { hyp / info }
2256 {
2257     #2 .code:n =
2258     {
2259         \tl_set:Nel__hyp_tmpa_tl {\text_expand:n{##1}}
2260         \__hyp_store_metadata:eo {pdf\str_lowercase:n{##1}}{\l__hyp_tmpa_tl}
2261         \tl_if_blank:nTF {##1}
2262         {
2263             \pdfmanagement_remove:nn {Info}{#2}
2264         }
2265         {
2266             \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2267             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2268             {
2269                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2270             }
2271         }
2272     }
2273     ,unknown .code:n =
2274     {
2275         \__hyp_text_pdfstring_info:eN {##1}\l__hyp_tmpa_str
2276         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2277         {
2278             \exp_args:Nno

```

```

2279         \pdfmanagement_add:nne {Info}
2280         { \l_keys_key_str } {\l__hyp_tmpa_str}
2281     }
2282 }
2283 }
2284 }
2285 \__hyp_setup_info_key:nn {author} {Author}
2286 \__hyp_setup_info_key:nn {title} {Title}
2287 \__hyp_setup_info_key:nn {producer} {Producer}
2288 \__hyp_setup_info_key:nn {creator} {Creator}
2289 % ignored key: addtopdfcreator
2290 \__hyp_setup_info_key:nn {subject} {Subject}
2291 \__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.

```

pdfmetadate (setup key) 2292 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2293 {
2294     \keys_define:nn { hyp }
2295     {
2296         pdf#1 .code:n =
2297         {
2298             \tl_if_blank:nTF {##1}
2299             {
2300                 \pdfmanagement_remove:nn {Info}{#2}
2301             }
2302             {
2303                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2304             }
2305             \__hyp_store_metadata:nn {pdf#1}{##1}
2306             \AddToDocumentProperties[document]{#1}{##1}
2307         }
2308     }
2309     \keys_define:nn { hyp / info }
2310     {
2311         #2 .code:n =
2312         {
2313             \tl_if_blank:nTF {##1}
2314             {
2315                 \pdfmanagement_remove:nn {Info}{#2}
2316             }
2317             {
2318                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2319             }
2320             \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2321         }
2322     }
2323 }
2324
2325 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2326 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2327 \keys_define:nn { hyp }
2328 {
2329     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }

```

```
2330     }
```

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

```
2331 \keys_define:nn { hyp }
2332 {
2333   ,pdftrapped .code:n =
2334   {
2335     \exp_args:Nne
2336     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2337   }
2338   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2339   {
2340     \pdfmanagement_add:nne {Info}{Trapped}
2341     {/
2342       \str_uppercase:f { \str_head:n { #1 } }
2343       \str_lowercase:f { \str_tail:n { #1 } }
2344     }
2345     \__hyp_store_metadata:ne {pdftrapped}
2346     {
2347       \str_uppercase:f { \str_head:n { #1 } }
2348       \str_lowercase:f { \str_tail:n { #1 } }
2349     }
2350   }
2351   ,_pdftrapped / unknown .code:n =
2352   {
2353     \msg_warning:nneee { hyp } { unknown-choice }
2354     { pdftrapped }
2355     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2356     { \exp_not:n {#1} }
2357   }
2358 }
```

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

```
2359 \keys_define:nn { hyp }
2360 {
2361   pdfinfo .code:n =
2362   {
2363     \keys_set:nn { hyp / info } { #1 }
2364   }
2365 }
```

Now we set some default values

```
2366 \keys_set:nn { hyp} {pdfcreator = LaTeX-with-hyperref}
2367 \keys_set:nn { hyp} {pdfauthor = }
2368 \keys_set:nn { hyp} {pdftitle = }
2369 \keys_set:nn { hyp} {pdfsubject = }
```

## 12.4 hyperxmp keys

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

```
2370 \clist_map_inline:nn
2371 {
```

```

2372 ,pdfcopyright
2373 ,pdftype
2374 ,pdflicenseurl
2375 ,pdfauthortitle
2376 ,pdfcaptionwriter
2377 ,pdfmetalang
2378 ,pdfsource
2379 ,pdfdocumentid
2380 ,pdfinstanceid
2381 ,pdfversionid
2382 ,pdfrendition
2383 ,pdfpublication
2384 ,pdfpubtype
2385 ,pdfbytes
2386 ,pdfnumpages
2387 ,pdfissn
2388 ,pdfeissn
2389 ,pdfisbn
2390 ,pdfbookedition
2391 ,pdfpublisher
2392 ,pdfvolumenum
2393 ,pdfissuenum
2394 ,pdfpagerange
2395 ,pdfdoi
2396 ,pdfurl
2397 ,pdfidentifier
2398 ,pdfsubtitle
2399 ,pdfpubstatus
2400 ,pdfcontactaddress
2401 ,pdfcontactcity
2402 ,pdfcontactregion
2403 ,pdfcontactpostcode
2404 ,pdfcontactcountry
2405 ,pdfcontactphone
2406 ,pdfcontactemail
2407 ,pdfcontacturl
2408 ,pdfdate
2409 }
2410 {
2411   \keys_define:nn { hyp }
2412   {
2413     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2414   }
2415 }
2416

```

## 12.5 Transitions

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

```

2417 \keys_define:nn { hyp }
2418 {
2419   pdfpageduration .code:n =
2420   {
2421     \tl_if_blank:nTF { #1 }

```

```

2422     {
2423     \pdfmanagement_remove:nn {Page}{Dur}
2424     }
2425     {
2426     \pdfmanagement_add:nnn {Page}{Dur}{#1}
2427     }
2428   }
2429 }

```

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

```

2430 \keys_define:nn { hyp }
2431 {
2432   pdfpagetransition .code:n =
2433   {
2434     \tl_if_blank:nTF {#1}
2435     {
2436       \pdfmanagement_remove:nn {Page}{Trans}
2437     }
2438     {
2439       \group_begin:
2440       \keys_set:nn { hyp / trans }{style=R,#1}
2441       \pdf_object_unnamed_write:ne { dict }
2442       {
2443         \pdfdict_use:n {l__hyp_page/Trans}
2444       }
2445       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}

```

```

2446         \group_end:
2447     }
2448 }
2449 }
2450 \keys_define:nn { hyp / trans }
2451 {
2452     ,style .choices:nn =
2453     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2454     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2455     ,style / unknown .code:n =
2456     {
2457         \msg_warning:nnee { hyp } { unknown-choice }
2458         { trans / style }
2459         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2460         { \exp_not:n {#1} }
2461     }
2462     ,duration .code:n =
2463     {
2464         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2465     }
2466     ,direction .choices:nn =
2467     { H,V }
2468     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2469     ,direction .choices:nn =
2470     { 0,90,180,270,315 }
2471     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2472     ,direction / None .code:n =
2473     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2474     ,direction / unknown .code:n =
2475     {
2476         \msg_warning:nnee { hyp } { unknown-choice }
2477         { trans / direction }
2478         {
2479             H~(horizontal,~only~Split,~Blinds),
2480             V~(vertical,~only~Split,~Blinds),
2481             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2482             90~(bottom~to~top,~only~Wipe),
2483             180~(right~to~left,~only~Wipe),
2484             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2485             315~(top~left~to~bottom,~only~Glitter),
2486             None~(only~Fly)
2487         }
2488         { \exp_not:n {#1} }
2489     }
2490     ,motion .choices:nn =
2491     { I,0 }
2492     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2493     ,motion / unknown .code:n =
2494     {
2495         \msg_warning:nnee { hyp } { unknown-choice }
2496         { trans / motion }
2497         { I~(inwards) , 0~(outwards) }
2498         { \exp_not:n {#1} }
2499     }

```

```

2500 ,scale .code:n =
2501   { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2502 ,opaque .choices:nn = {true,false}
2503   { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1} }
2504 ,opaque / unknown .code:n =
2505   {
2506     \msg_warning:nnee { hyp } { unknown-choice }
2507     { trans / B }
2508     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2509     { \exp_not:n {#1} }
2510   }
2511 % try to set unknown keys as style
2512 ,unknown .code:n =
2513   {
2514     % warning ...
2515     \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2516   }
2517 }

```

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

```

2518 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
  Unfinished Form field code
2519 \NewDocumentCommand \MakeFieldObject { m m }
2520   {
2521     \pdfxform_new:nnn { #2 }{ } { #1 }
2522   }
2523
2524
2525 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2526 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2527
2528 \let\HyField@afields\@empty
2529 \let\HyField@cofields\@empty
2530 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2531
2532 % the value doesn't matter, but with a prop we avoid duplicates and it is
2533 % clearly faster than removing them from a sequence
2534 \def\HyField@AuxAddToFields#1
2535   {
2536     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2537   }%
2538
2539 %fields with empty key get a value too -- lets hope that
2540 %this give the expected behaviour
2541 \def\HyField@AuxAddToCoFields #1 #2
2542   {
2543     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2544   }
2545
2546 \Hy@AtBeginDocument
2547   {
2548     \if@filesw
2549       \immediate\write\@mainaux{%
2550         \string\providecommand\string\HyField@AuxAddToFields[1]{}%

```

```

2551     }%
2552     \immediate\write\@mainaux{%
2553       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2554     }%
2555   \fi
2556   \let\HyField@AfterAuxOpen\@firstofone
2557 }%
2558
2559 \def\HyField@AddToFields
2560 {
2561   \exp_args:N\HyField__hypAddToFields
2562   {
2563     \pdfannot_box_ref_last:
2564   }
2565   \ifx\Fld@calculate@code\@empty
2566   \else
2567     \begingroup
2568     \Hy@safe@activetrue
2569     \edef\Hy@temp{%
2570       \endgroup
2571       \if@filesw
2572         \write\@mainaux
2573         {
2574           \string\HyField@AuxAddToCoFields
2575           {
2576             \Fld@calculate@sortkey
2577           }
2578           {
2579             \pdfannot_box_ref_last:
2580           }
2581         }
2582       \fi
2583     }%
2584     \Hy@temp
2585   \fi
2586 }%
2587
2588 \def\HyField__hypAddToFields#1{
2589   \HyField@AfterAuxOpen{%
2590     \if@filesw
2591       \write\@mainaux{%
2592         \string\HyField@AuxAddToFields{#1}%
2593       }%
2594     \fi
2595   }%
2596 }%
2597
2598 \ExplSyntaxOff
2599 \ExplSyntaxOn
2600
2601 \def\@Form[#1]
2602 {
2603   \kvsetkeys{Form}{#1}
2604   \pdf@ifdraftmode{}

```

```

2605 {
2606   \Hy@FormObjects
2607   \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2608   {
2609     \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2610     %\pdfmanagement_show:n { Catalog / AcroForm }
2611   }
2612   \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2613   {
2614     \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2615     {
2616       \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2617     }
2618     \seq_sort:Nn \l__hyp_tmpa_seq
2619     {
2620       \str_compare:nNnTF {##1} > {##2}
2621       { \sort_return_swapped: }
2622       { \sort_return_same: }
2623     }
2624     \seq_map_inline:Nn \l__hyp_tmpa_seq
2625     {
2626       \pdfmanagement_add:nne { Catalog / AcroForm }
2627       { CO }
2628       {
2629         \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2630       }
2631     }
2632   }
2633   \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2634   {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2635   \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2636   {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2637   \pdfmanagement_add:nne {Catalog /AcroForm}
2638   {DA}{(/Helv~10~Tf~0~g)}
2639   \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2640   {
2641     \legacy_if:nT { HyField@NeedAppearances }
2642     {
2643       \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2644     }
2645   }
2646   {
2647     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2648   }
2649 }
2650 }
2651 \ExplSyntaxOff
2652 \let\@endForm\@empty
2653 \let\HyAnn@AbsPageLabel\@empty
2654 \let\Fld@pageobjref\@empty
2655
2656 \ExplSyntaxOn
2657 \newcount\HyAnn@Count
2658 \HyAnn@Count=\z@

```

```

2659 \def\HyAnn@AbsPageLabel
2660 {
2661   \global\advance\HyAnn@Count by\@ne
2662   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2663   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2664   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2665   \_hyp\_property\_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2666   \property\_ref\_undefined\_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2667 }%
2668 \prg\_generate\_conditional\_variant:Nnn \property\_if\_recorded:mn {ee} {T}
2669 \def\Fld@pageobjref
2670 {
2671   \property\_if\_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2672   {
2673     /P~\pdf\_pageobject\_ref:e
2674     {
2675       \property\_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2676     }
2677   }
2678 }
2679 \ExplSyntaxOff
2680 \ExplSyntaxOn
2681 %% check if the attr should be set through
2682 %% hooks.
2683 %% check if options are missing.
2684 \def\@TextField[#1]#2{% parameters, label
2685   \def\Fld@name{#2}%
2686   \let\Fld@default\@empty
2687   \let\Fld@value\@empty
2688   \def\Fld@width{\DefaultWidthofText}%
2689   \def\Fld@height{%
2690     \ifFld@multiline
2691       \DefaultHeightofTextMultiline
2692     \else
2693       \DefaultHeightofText
2694     \fi
2695 }%
2696 \begingroup
2697   \expandafter\HyField@SetKeys\expandafter{%
2698     \DefaultOptionsofText,#1%
2699   }%
2700   \PDFForm@Name
2701   \HyField@FlagsText
2702   \ifFld@hidden\def\Fld@width{1sp}\fi
2703   \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2704   \LayoutTextField{#2}{%
2705     \leavevmode
2706     \HyAnn@AbsPageLabel
2707     \Hy@escapeform\PDFForm@Text
2708     \pdfannot\_box:nmmn
2709       {\Fld@width}
2710       {\Fld@height}
2711       {Opt} %is this correct?
2712       {\PDFForm@Text}

```

```

2713     \MakeTextField{\Fld@width}{\Fld@height}
2714     \HyField@AddToFields
2715   }%
2716 \endgroup
2717 }
2718 \providecommand\@curropt{}
2719 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2720   \def\Fld@name{#2}
2721   \let\Fld@default\relax
2722   \let\Fld@value\relax
2723   \def\Fld@width{\DefaultWidthofChoiceMenu}
2724   \def\Fld@height{\DefaultHeightofChoiceMenu}
2725   \begingroup
2726     \Fld@menulength=0 %
2727     \@tempdima\z@
2728     \clist_map_variable:nNn { #3 } \@curropt
2729     %\@for\@curropt:=#3\do
2730     {%
2731       \expandafter\Fld@checkequals\@curropt==\%
2732       \Hy@StepCount\Fld@menulength
2733       \settowidth{\@tempdimb}{\@currDisplay}%
2734       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2735     }%
2736     \advance\@tempdima by~15\p@
2737     \begingroup
2738       \HyField@SetKeys{#1}
2739     \edef\x{\endgroup
2740       \noexpand\expandafter
2741       \noexpand\HyField@SetKeys
2742       \noexpand\expandafter{%
2743         \expandafter\noexpand\csgname DefaultOptionsof%
2744         \ifFld@radio
2745           Radio%
2746         \else
2747           \ifFld@combo
2748             \ifFld@popdown
2749               PopdownBox%
2750             \else
2751               ComboBox%
2752             \fi
2753           \else
2754             ListBox%
2755           \fi
2756         \fi
2757       \endcsname
2758     }%
2759   }\x
2760   \HyField@SetKeys{#1}%
2761   \PDFForm@Name
2762   \ifFld@hidden\def\Fld@width{1sp}\fi
2763   \ifx\Fld@value\relax
2764     \let\Fld@value\Fld@default
2765   \fi
2766   \LayoutChoiceField{#2}{%

```

```

2767 \ifFld@radio
2768 \HyField@FlagsRadioButton
2769 \__hypRadio{#3}%
2770 \else
2771 \beginingroup
2772 \HyField@FlagsChoice
2773 \ifdim\Fld@width<\@tempdima
2774 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2775 \edef\Fld@width{\the\@tempdima}%
2776 \fi
2777 \ifFld@combo
2778 \else
2779 \@tempdima=\the\Fld@menulength\Fld@charsize
2780 \advance\@tempdima by-\Fld@borderwidth bp %
2781 \advance\@tempdima by-\Fld@borderwidth bp %
2782 \edef\Fld@height{\the\@tempdima}%
2783 \fi
2784 \__hypListBox{#3}%
2785 \endgroup
2786 \fi
2787 }%
2788 \endgroup
2789 }
2790
2791 \def\__hypRadio#1{%
2792 \Fld@listcount=0-%
2793 %\show\Fld@default
2794 \EdefEscapeName\Fld@default{\Fld@default}%
2795 \clist_map_variable:nNn { #1 } \@curropt
2796 %\@for\@curropt:=#1\do
2797 {%
2798 \expandafter\Fld@checkequals\@curropt==\%
2799 \EdefEscapeName\@currValue{\@currValue}%
2800 \Hy@StepCount\Fld@listcount
2801 \@currDisplay\space
2802 \leavevmode
2803 \HyAnn@AbsPageLabel
2804 \Hy@escapeform\PDFForm@Radio
2805 \pdfxform_if_exist:nF { __hyp_xform_Ding }
2806 {
2807 \pdfxform_new:nnn { __hyp_xform_Ding } {}
2808 {
2809 \group_begin:
2810 \fontfamily{pzd}
2811 \fontencoding{U}
2812 \fontseries{m}
2813 \fontshape{n}
2814 \selectfont
2815 \char123
2816 \group_end:
2817 }
2818 }
2819 \pdfannot_box:nnne
2820 {\Fld@width}

```

```

2821     {\Fld@height}
2822     {Opt} %is this correct?
2823     {
2824         \PDFForm@Radio
2825         /AP
2826         <<
2827         /N
2828         <<
2829         /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2830         %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2831         >>
2832         >>
2833     }
2834     {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2835     \int_compare:nNnT { \Fld@listcount} = { 1 }
2836     { \HyField@AddToFields }
2837     \c_space_tl % deliberate space between radio buttons
2838         % to do: --> should be configurable
2839 }%
2840 }
2841
2842 \newcount\Fld@listcount
2843 \def\__hypListBox#1
2844 {
2845     \HyField@PDFChoices{#1}
2846     \mode_leave_vertical:
2847     \HyAnn@AbsPageLabel
2848     \Hy@escapeform\PDFForm@List
2849     \pdf_link_user:nnn
2850         {widget} %perhaps we need more types??
2851         {\PDFForm@List}
2852         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2853     \HyField@AddToFields
2854 }
2855
2856
2857 \def\@PushButton[#1]#2{% parameters, label
2858     \def\Fld@name{#2}%
2859     \group_begin:
2860         \exp_args:No\HyField@SetKeys
2861         {
2862             \DefaultOptionsofPushButton,#1
2863         }
2864         \PDFForm@Name
2865         \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2866         {
2867             \HyField@FlagsPushButton
2868             \legacy_if:nT {Fld@hidden}
2869             {
2870                 \def\Fld@width{1sp}
2871             }
2872             \LayoutPushButtonField
2873             {
2874                 \mode_leave_vertical:

```

```

2875         \HyAnn@AbsPageLabel
2876         \Hy@escapeform\PDFForm@Push
2877         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2878         \pdfannot_box:nnnn
2879             {\box_wd:N\l_tmpa_box}
2880             {\box_ht:N\l_tmpa_box}
2881             {\box_dp:N\l_tmpa_box} %is this correct?
2882             {\PDFForm@Push}
2883             {\box_use:N\l_tmpa_box}
2884         \HyField@AddToFields
2885     }
2886 }
2887 {
2888     \msg_error:nn { hyp }{ pdfa-no-push-button }
2889     \LayoutPushButtonField
2890     {
2891         \mode_leave_vertical:
2892         \MakeButtonField{#2}
2893     }
2894 }
2895 \group_end:
2896 }
2897
2898 \def\@Submit[#1]#2
2899 {
2900     \def\Fld@width {\DefaultWidthofSubmit}
2901     \def\Fld@height{\DefaultHeightofSubmit}
2902     \group_begin:
2903         \exp_args:No\HyField@SetKeys
2904         {
2905             \DefaultOptionsofSubmit,#1
2906         }
2907     \HyField@FlagsPushButton
2908     \HyField@FlagsSubmit
2909     \legacy_if:nT { Fld@hidden }
2910     {
2911         \def\Fld@width{1sp}
2912     }
2913     \mode_leave_vertical:
2914     \HyAnn@AbsPageLabel
2915     \Hy@escapeform\PDFForm@Submit
2916     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2917     \pdfxform_if_exist:nF
2918     { __hyp_xform_Submit }
2919     {
2920         \pdfxform_new:nnn { __hyp_xform_Submit }{}
2921         {
2922             \fbox{\color_select:n{yellow}\textsf{Submit}}
2923         }
2924         \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2925         {
2926             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2927         }
2928     }

```

```

2929 \pdfannot_box:nnnn
2930   {\box_wd:N\l_tmpa_box}
2931   {\box_ht:N\l_tmpa_box}
2932   {\box_dp:N\l_tmpa_box} %is this correct?
2933   {
2934     \PDFForm@Submit
2935     /AP<<
2936       /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2937       /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2938     >>
2939   }
2940   \HyField@AddToFields
2941   \box_use:N\l_tmpa_box
2942
2943 \group_end:
2944 }
2945
2946 \def\@Reset[#1]#2
2947 {
2948   \def\Fld@width {\DefaultWidthofReset}
2949   \def\Fld@height{\DefaultHeightofReset}
2950   \group_begin:
2951     \exp_args:No\HyField@SetKeys
2952     {
2953       \DefaultOptionsofReset,#1
2954     }
2955     \mode_leave_vertical:
2956     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2957     {
2958       \HyField@FlagsPushButton
2959       \legacy_if:nT { Fld@hidden }
2960         { \def\Fld@width{1sp} }
2961       \HyAnn@AbsPageLabel
2962       \Hy@escapeform\PDFForm@Reset
2963       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2964       \pdfannot_box:nnnn
2965         {\box_wd:N\l_tmpa_box}
2966         {\box_ht:N\l_tmpa_box}
2967         {\box_dp:N\l_tmpa_box} %is this correct?
2968         { \PDFForm@Reset }
2969       \HyField@AddToFields
2970       \box_use:N \l_tmpa_box
2971     }
2972     {
2973       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2974       \MakeButtonField{#2}
2975     }
2976   \group_end:
2977 }
2978
2979 \def\@CheckBox[#1]#2
2980 {% parameters, label
2981   \def\Fld@name{#2}
2982   \def\Fld@default{0}

```

```

2983 \group_begin:
2984 \def\Fld@width {\DefaultWidthofCheckBox}
2985 \def\Fld@height{\DefaultHeightofCheckBox}
2986 \exp_args:No\HyField@SetKeys
2987 {
2988   \DefaultOptionsofCheckBox,#1
2989 }
2990 \PDFForm@Name
2991 \HyField@FlagsCheckBox
2992 \legacy_if:nT { Fld@hidden }
2993 {
2994   \def\Fld@width{1sp}
2995 }
2996 \LayoutCheckField{#2}
2997 {
2998   \mode_leave_vertical:
2999   \HyAnn@AbsPageLabel
3000   \Hy@escapeform\PDFForm@Check
3001   \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3002   {
3003     \pdfxform_new:nnn
3004     {__hyp_xform_CheckMarkYes}{}
3005     {
3006       \group_begin:
3007       \fontfamily{pzd}
3008       \fontencoding{U}
3009       \fontseries{m}
3010       \fontshape{n}
3011       \selectfont
3012       \char51
3013       \group_end:
3014     }
3015     \pdfxform_new:nnn
3016     {__hyp_xform_CheckMarkOff}{}
3017     {
3018       \group_begin:
3019       \fontfamily{pzd}
3020       \fontencoding{U}
3021       \fontseries{m}
3022       \fontshape{n}
3023       \selectfont
3024       \phantom{\char51} %perhaps xetex needs some small glyph ..
3025       \group_end:
3026     }
3027   }
3028   \pdfannot_box:nmm
3029   {\Fld@width}
3030   {\Fld@height}
3031   {Opt} %is this correct?
3032   {\PDFForm@Check}
3033   \HyField@AddToFields %check if this works with xelatex ...
3034 }
3035 \group_end:
3036 }

```

```

3037 \ExplSyntaxOff
3038
3039 %hm. Should a luatex driver use type1 fonts in fields????
3040 \ExplSyntaxOn
3041 \def\Hy@FormObjects
3042 {
3043   \pdf_object_new:n   {__hyp/Encoding/pdfdoc }
3044   \pdf_object_new:n   {__hyp/Font/ZaDb }
3045   \pdf_object_new:n   {__hyp/Font/Helv }
3046   \pdf_object_write:mne {__hyp/Encoding/pdfdoc } { dict }
3047   {
3048     /Type/Encoding
3049     /Differences[
3050       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3051       /ring/tilde
3052       \c_space_tl
3053       39/quotesingle
3054       \c_space_tl
3055       96/grave %
3056       \iow_newline:
3057       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3058       /fraction/guilsinglleft/guilsinglright/minus/perthousand
3059       /quotedblbase/quotedblleft/quotedblright/quoteleft
3060       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3061       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3062       \iow_newline:
3063       164/currency
3064       \c_space_tl
3065       166/brokenbar
3066       \c_space_tl
3067       168/dieresis/copyright/ordfeminine
3068       \c_space_tl
3069       172/logicalnot/.notdef/registered/macron/degree/plusminus
3070       /twosuperior/threesuperior/acute/mu
3071       \c_space_tl
3072       183/periodcentered/cedilla/onesuperior/ordmasculine
3073       \c_space_tl
3074       188/onequarter/onehalf/threequarters
3075       \iow_newline:
3076       192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3077       /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3078       /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3079       /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3080       /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3081       /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3082       /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3083       /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
3084       /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3085       /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3086     ]
3087   }
3088   \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3089   {
3090     /Type/Font

```

```

3091         /Subtype/Type1
3092         /Name/ZaDb
3093         /BaseFont/ZapfDingbats
3094     }
3095     \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3096     {
3097         /Type/Font
3098         /Subtype/Type1
3099         /Name/Helv
3100         /BaseFont/Helvetica
3101         /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3102     }
3103     \global\let\Hy@FormObjects\relax
3104 }
3105 \ExplSyntaxOff
3106 \providecommand*{\Fld@pageobjref}{}
3107 \ifcsname pdf@escapestring\endcsname
3108     \def\Hy@escapeform#1{%
3109         \ifHy@pdfescapeform
3110             \let\Hy@escapestring\pdfescapestring
3111         \else
3112             \let\Hy@escapestring\@firstofone
3113         \fi
3114     }%
3115     \Hy@escapeform{}%
3116 \else
3117     \let\Hy@escapestring\@firstofone
3118     \def\Hy@escapeform#1{%
3119         \ifHy@pdfescapeform
3120             \def\Hy@escapestring##1{%
3121                 \noexpand\Hy@escapestring{\noexpand##1}%
3122             }%
3123             \edef\Hy@temp{#1}%
3124             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring}\@nil
3125         \def\Hy@escapestring##1{%
3126             \@ifundefined{Hy@esc@\string##1}{%
3127                 ##1%
3128                 \ThisShouldNotHappen
3129             }{%
3130                 \csname Hy@esc@\string##1\endcsname
3131             }%
3132         }%
3133     \else
3134         \let\Hy@escapestring\@firstofone
3135     \fi
3136 }%
3137 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3138     \ifx\#3\%
3139     \else
3140         \expandafter
3141         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3142         \Hy@ReturnAfterFi{%
3143             \Hy__hypescapeform#3\@nil
3144         }%

```

```

3145     \fi
3146   }%
3147   \fi
3148   \def\PDFForm@Name{%
3149     \PDFForm__hypName\Fld@name
3150     \ifx\Fld@altname\relax
3151     \else
3152       \PDFForm__hypName\Fld@altname
3153     \fi
3154     \ifx\Fld@mappingname\relax
3155     \else
3156       \PDFForm__hypName\Fld@mappingname
3157     \fi
3158   }
3159   \def\PDFForm__hypName#1{%
3160     \begingroup
3161       \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3162       \ifHy@unicode
3163         \Hy@unicodedefalse
3164       \fi
3165       \fi
3166       \pdfstringdef\Hy@gtemp#1%
3167     \endgroup
3168     \let#1\Hy@gtemp
3169   }
3170   \def\Fld@X@additionalactions{%
3171     \ifx\Fld@keystroke@code\@empty
3172     \else
3173       /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3174     \fi
3175     \ifx\Fld@format@code\@empty
3176     \else
3177       /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3178     \fi
3179     \ifx\Fld@validate@code\@empty
3180     \else
3181       /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3182     \fi
3183     \ifx\Fld@calculate@code\@empty
3184     \else
3185       /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3186     \fi
3187     \ifx\Fld@onfocus@code\@empty
3188     \else
3189       /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3190     \fi
3191     \ifx\Fld@onblur@code\@empty
3192     \else
3193       /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3194     \fi
3195     \ifx\Fld@onmousedown@code\@empty
3196     \else
3197       /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3198     \fi

```

```

3199 \ifx\Fld@onmouseup@code\@empty
3200 \else
3201 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3202 \fi
3203 \ifx\Fld@onenter@code\@empty
3204 \else
3205 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3206 \fi
3207 \ifx\Fld@onexit@code\@empty
3208 \else
3209 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3210 \fi
3211 }
3212 \ExplSyntaxOn
3213 \def\Fld@additionalactions
3214 {%
3215 \exp_args:Ne\str_if_eq:nF {\Fld@X@additionalactions}{}
3216 {
3217 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3218 {/AA<<\Fld@X@additionalactions>>}
3219 }
3220 }
3221 \ExplSyntaxOff
3222 \def\Fld@annotnames{%
3223 /T(\Fld@name)%
3224 \ifx\Fld@altname\relax
3225 \else
3226 /TU(\Fld@altname)%
3227 \fi
3228 \ifx\Fld@mappingname\relax
3229 \else
3230 /TM(\Fld@mappingname)%
3231 \fi
3232 }
3233 \ExplSyntaxOn
3234 \def\PDFForm@Check
3235 {
3236 /Subtype/Widget
3237 ~\Fld@annotflags
3238 ~\Fld@pageobjref
3239 ~\Fld@annotnames
3240 /FT/Btn
3241 \Fld@flags
3242 /Q~\Fld@align
3243 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3244 /AP
3245 <<
3246 /N
3247 <<
3248 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3249 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3250 >>
3251 >>
3252 /MK<<

```

```

3253 \int_compare:nNnF {\Fld@rotation}={0}
3254 {
3255   /R~\Fld@rotation
3256 }
3257 \tl_if_empty:NF\Fld@bordercolor
3258 {
3259   /BC[\Fld@bordercolor]
3260 }
3261 \tl_if_empty:NF\Fld@bcolor
3262 {
3263   /BG[\Fld@bcolor]
3264 }
3265 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3266 >>
3267 /DA
3268 (
3269   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3270   \tl_if_empty:NF \Fld@color
3271   {
3272     \c_space_tl \Fld@color
3273   }
3274 )
3275 /H/P
3276 \legacy_if:nTF {Fld@checked}
3277 {
3278   /V/Yes /AS/Yes
3279 }
3280 {
3281   /V/Off /AS/Off
3282 }
3283 \Fld@additionalactions
3284 }
3285 \ExplSyntaxOff
3286 \ExplSyntaxOn
3287 \def\PDFForm@Push
3288 {
3289   /Subtype/Widget
3290   ~\Fld@annotflags
3291   ~\Fld@pageobjref
3292   ~\Fld@annotnames
3293   /FT/Btn
3294   ~\Fld@flags
3295   /H/P
3296   /BS<<W~\Fld@borderwidth/S/\Fld@borderstyle>>
3297   \bool_if:nT
3298   {
3299     !\int_compare_p:nNn {\Fld@rotation} = {0}
3300     ||
3301     \tl_if_exist_p:N \Fld@bordercolor
3302   }
3303   {
3304     /MK
3305     <<
3306     \int_compare:nNnF {\Fld@rotation} = {0}

```

```

3307         {
3308             /R~\Fld@rotation
3309         }
3310         \tl_if_exist:NT \Fld@bordercolor
3311         {
3312             /BC[\Fld@bordercolor]
3313         }
3314     >>
3315 }
3316 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3317 \Fld@additionalactions
3318 }
3319
3320 \ExplSyntaxOff
3321 \def\PDFForm@List{%
3322     /Subtype/Widget%
3323     \Fld@annotflags
3324     \Fld@pageobjref
3325     \Fld@annotnames
3326     /FT/Ch%
3327     \Fld@flags
3328     /Q \Fld@align
3329     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3330     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3331         \ifx\Fld@bordercolor\relax\else 1\fi
3332         \ifx\Fld@bcolor\relax \else 1\fi
3333     \space
3334 \else
3335     /MK<<%
3336         \ifnum\Fld@rotation=\z@
3337         \else
3338             /R \Fld@rotation
3339         \fi
3340         \ifx\Fld@bordercolor\relax
3341         \else
3342             /BC[\Fld@bordercolor]%
3343         \fi
3344         \ifx\Fld@bcolor\relax
3345         \else
3346             /BG[\Fld@bcolor]%
3347         \fi
3348     >>%
3349 \fi
3350 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3351     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3352 \Fld@choices
3353 \Fld@additionalactions
3354 }
3355 \ExplSyntaxOn
3356 \def\PDFForm@Radio
3357 {
3358     /Subtype/Widget
3359     ~\Fld@annotflags
3360     ~\Fld@pageobjref

```

```

3361 ~\Fld@annotnames
3362 /FT/Btn
3363 \Fld@flags
3364 /H/P
3365 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3366 /MK<<
3367 \ifnum\Fld@rotation=\z@
3368 \else
3369 /R~\Fld@rotation
3370 \fi
3371 \ifx\Fld@bordercolor\relax
3372 \else
3373 /BC[\Fld@bordercolor]%
3374 \fi
3375 \ifx\Fld@bcolor\relax
3376 \else
3377 /BG[\Fld@bcolor]%
3378 \fi
3379 /CA(\Hy\escapestring{\Fld@radiosymbol})%
3380 >>
3381 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3382 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3383 \ifx\Fld@default\@empty
3384 /V/Off%
3385 /DV/Off%
3386 \else
3387 /V/\Fld@default
3388 /DV/\Fld@default
3389 \fi
3390 \Fld@additionalactions
3391 }
3392 \ExplSyntaxOff
3393 \ExplSyntaxOn
3394 % Does an appearance dict make sense here?
3395 \def\PDFForm@Text
3396 {
3397 /Subtype/Widget
3398 ~\Fld@annotflags
3399 ~\Fld@pageobjref
3400 ~\Fld@annotnames
3401 /FT/Tx
3402 ~\Fld@flags
3403 /Q~\Fld@align
3404 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3405 \bool_if:nT
3406 {
3407 !\int_compare_p:nNn {\Fld@rotation} = {0}
3408 ||
3409 \tl_if_exist_p:N \Fld@bordercolor
3410 ||
3411 \tl_if_exist_p:N \Fld@bcolor
3412 }
3413 {
3414 /MK

```

```

3415     <<
3416         \int_compare:nNnF {\Fld@rotation} = {0}
3417         {
3418             /R~\Fld@rotation
3419         }
3420     \tl_if_exist:NT \Fld@bordercolor
3421     {
3422         /BC[\Fld@bordercolor]
3423     }
3424     \tl_if_exist:NT \Fld@bcolor
3425     {
3426         /BG[\Fld@bcolor]
3427     }
3428     >>
3429 }
3430 /DA
3431 (
3432     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3433     \tl_if_empty:NF {\c_space_tl\Fld@color}
3434 )
3435 /DV(\Hy@escapestring{\Fld@default})
3436 /V(\Hy@escapestring{\Fld@value})
3437 ~\Fld@additionalactions
3438 \int_compare:nNnT { \Fld@maxlen}>{0}
3439 {
3440     /MaxLen~\Fld@maxlen
3441 }
3442 }
3443 \ExplSyntaxOff
3444
3445 \def\PDFForm@Submit{%
3446     /Subtype/Widget%
3447     \Fld@annotflags
3448     \Fld@pageobjref
3449     \Fld@annotnames
3450     /FT/Btn%
3451     \Fld@flags
3452     /H/P%
3453     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3454     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3455         \ifx\Fld@bordercolor\relax\else 1\fi
3456         \space
3457     \else
3458         /MK<<%
3459         \ifnum\Fld@rotation=\z@
3460         \else
3461             /R \Fld@rotation
3462         \fi
3463         \ifx\Fld@bordercolor\relax
3464         \else
3465             /BC[\Fld@bordercolor]%
3466         \fi
3467     >>%
3468     \fi

```

```

3469 /A<<%
3470 /S/SubmitForm%
3471 /F<<%
3472 /FS/URL%
3473 /F(\Hy@escapestring{\Form@action})%
3474 >>%
3475 \Fld@submitflags
3476 >>%
3477 \Fld@additionalactions
3478 }
3479 \ExplSyntaxOn
3480 \def\PDFForm@Reset{%
3481 /Subtype/Widget%
3482 \Fld@annotflags
3483 \Fld@pageobjref
3484 \Fld@annotnames
3485 /FT/Btn%
3486 \Fld@flags
3487 /H/P%
3488 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3489 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3490 \ifx\Fld@bordercolor\relax\else 1\fi
3491 \space
3492 \else
3493 /MK<<%
3494 \ifnum\Fld@rotation=\z@
3495 \else
3496 /R~\Fld@rotation
3497 \fi
3498 \ifx\Fld@bordercolor\relax
3499 \else
3500 /BC[\Fld@bordercolor]%
3501 \fi
3502 >>%
3503 \fi
3504 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3505 /A<</S/ResetForm>>%
3506 \Fld@additionalactions
3507 }%
3508
3509
3510 %these patterns are used in hyperref checks.
3511 %it is unclear if they are really useful and if a backend support is
3512 %needed.
3513 \str_case:VnF \c_sys_backend_str
3514 {
3515 { pdfmode }
3516 {
3517 \def\HyPat@ObjRef
3518 {
3519 [0-9]*[1-9][0-9]*~0~R
3520 }
3521 }
3522 { dvipdfmx }

```

```

3523 {
3524   \def\HyPat@ObjRef
3525   {
3526     @[\~]+
3527   }
3528 }
3529 { xdvipdfmx }
3530 {
3531   \def\HyPat@ObjRef
3532   {
3533     @[\~]+
3534   }
3535 }
3536 }
3537 { %also set in hyperref sty, so probably not needed.
3538   \def\HyPat@ObjRef/{.+}
3539 }
3540
3541
3542 \ExplSyntaxOff
3543 % UF: removed Hy@writebookmark
3544 %   \Hy@currentbookmarklevel{0}
3545 %   \Hy@numberline
3546 %   \__hypwritetorep
3547 %   counter{bookmark@seq@number}
3548 % removed \HyPsd@SanitizeForOutFile, not needed
3549 % removed \currentpdfbookmark, defined by bookmark,
3550 % should use \newcommand there
3551 % removed \subpdfbookmark, defined by bookmark,
3552 % should use \newcommand there
3553 % removed \belowpdfbookmark, defined by bookmark,
3554 % should use \newcommand there
3555 % removed \pdfbookmark, defined by bookmark,
3556 % \BOOKMARK
3557 % \@BOOKMARK
3558 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3559 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3560 %% removed \ReadBookmarks / unneeded with bookmark.
3561 %% removed \Hy@OutlineName
3562 %% removed \check@bm@number
3563 %% removed \calc@bm@number
3564
3565 \ifHy@implicit
3566 \else
3567   \expandafter\endinput
3568 \fi
3569 \newlength\Hy@SectionHShift
3570 \def\Hy@SectionAnchorHref#1{%
3571   \ifx\protect\@typeset@protect
3572     \Hy__hypSectionAnchor{#1}%
3573   \fi
3574 }
3575 \DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
3576   \leavevmode

```

```

3577 \hbox to Opt{%
3578   \kern-\Hy@SectionHShift
3579   \Hy@raisedlink{%
3580     \hyper@anchorstart{#1}\hyper@anchorend
3581   }%
3582   \hss
3583 }%
3584 }
3585 \@ifundefined{hyper@nopatch@sectioning}
3586 {
3587   \let\H@old@ssect\@ssect
3588   \def\@ssect#1#2#3#4#5{%
3589     \Hy@MakeCurrentHrefAuto{section*}%
3590     \setlength{\Hy@SectionHShift}{#1}%
3591     \begingroup
3592       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3593       \toks\tw@\expandafter{%
3594         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3595         #5%
3596       }%
3597       \edef\x{\endgroup
3598         \the\toks@\the\toks\tw@}%
3599       }\x
3600   }
3601   \let\H@old@schapter\@schapter
3602   \def\@schapter#1{%
3603     \begingroup
3604       \let\@mkboth\@gobbletwo
3605       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3606       \Hy@raisedlink{%
3607         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3608       }%
3609     \endgroup
3610     \H@old@schapter{#1}%
3611   }
3612   \@ifundefined{@chapter}{-}{%
3613     \let\Hy@org@chapter\@chapter
3614     \def\@chapter{%
3615       \def\Hy@next{%
3616         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3617         \Hy@raisedlink{%
3618           \hyper@anchorstart{\@currentHref}\hyper@anchorend
3619         }%
3620       }%
3621       \ifnum\c@secnumdepth>\m@ne
3622         \@ifundefined{if@mainmatter}%
3623         \iftrue{\csname if@mainmatter\endcsname}%
3624         \let\Hy@next\relax
3625       \fi
3626     \fi
3627     \Hy@next
3628     \Hy@org@chapter
3629   }%
3630 }

```

```

3631 \let\H@old@part\@part
3632 \begingroup\expandafter\expandafter\expandafter\endgroup
3633 \expandafter\ifx\csname chapter\endcsname\relax
3634 \let\Hy@secnum@part\z@
3635 \else
3636 \let\Hy@secnum@part\m@ne
3637 \fi
3638 \def\@part{%
3639 \ifnum\Hy@secnum@part>\c@secnumdepth
3640 \phantomsection
3641 \fi
3642 \H@old@part
3643 }
3644 \let\H@old@spart\@spart
3645 \def\@spart#1{%
3646 \Hy@MakeCurrentHrefAuto{part*}%
3647 \Hy@raisedlink{%
3648 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3649 }%
3650 \H@old@spart{#1}%
3651 }
3652 \let\H@old@sect\@sect
3653 \def\@sect#1#2#3#4#5#6[#7]#8{%
3654 \ifnum #2>\c@secnumdepth
3655 \expandafter\@firstoftwo
3656 \else
3657 \expandafter\@secondoftwo
3658 \fi
3659 {%
3660 \Hy@MakeCurrentHrefAuto{section*}%
3661 \setlength{\Hy@SectionHShift}{#3}%
3662 \begingroup
3663 \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3664 \toks\tw@\expandafter{%
3665 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3666 #8%
3667 }%
3668 \edef\x{\endgroup
3669 \the\toks@\the\toks\tw@}%
3670 }\x
3671 }{%
3672 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]{#8}%
3673 }%
3674 }
3675 }{}
3676 \expandafter\def\csname Parent-4\endcsname{}
3677 \expandafter\def\csname Parent-3\endcsname{}
3678 \expandafter\def\csname Parent-2\endcsname{}
3679 \expandafter\def\csname Parent-1\endcsname{}
3680 \expandafter\def\csname Parent0\endcsname{}
3681 \expandafter\def\csname Parent1\endcsname{}
3682 \expandafter\def\csname Parent2\endcsname{}
3683 \expandafter\def\csname Parent3\endcsname{}
3684 \expandafter\def\csname Parent4\endcsname{}

```

```

3685 %%
3686 %% End of file 'hgeneric-testphase.def'.
3687 </package>
3688 <*colorscheme>
3689 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3690 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3691 % linkcolor=
3692 %,filecolor=
3693 %,urlcolor=
3694 %,menucolor=
3695 %,runcolor=
3696 %,linkbordercolor=
3697 %,filebordercolor=
3698 %,urlbordercolor=
3699 %,menubordercolor=
3700 %,runbordercolor=
3701
3702 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3703 {
3704   linkcolor      = [rgb]{1,0,0}, %red
3705   filecolor      = [rgb]{0,1,1}, %cyan
3706   urlcolor       = [rgb]{1,0,1}, %magenta
3707   menucolor      = [rgb]{1, 0, 0}, %red
3708   runcolor       = [rgb]{0,1,1}, %cyan
3709   %-----
3710   linkbordercolor = [rgb]{1, 0 ,0 },
3711   filebordercolor = [rgb]{0, .5, .5},
3712   urlbordercolor  = [rgb]{0, 1, 1},
3713   menubordercolor = [rgb]{1, 0, 0},
3714   runbordercolor  = [rgb]{0, .7, .7}
3715 }
3716
3717 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3718 {
3719   linkcolor      = [rgb]{0,0.2,0.6},
3720   filecolor      = [rgb]{0.8,0,0.8},
3721   urlcolor       = [rgb]{0.8,0,0.8},
3722   menucolor      = [rgb]{0,0.2,0.6},
3723   runcolor       = [rgb]{0.8,0,0.8},
3724   %----- %-----
3725   linkbordercolor = [rgb]{0,0.2,0.6},
3726   filebordercolor = [rgb]{0.8,0,0.8},
3727   urlbordercolor  = [rgb]{0.8,0,0.8},
3728   menubordercolor = [rgb]{0,0.2,0.6},
3729   runbordercolor  = [rgb]{0.8,0,0.8}
3730 }
3731
3732 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3733 { %two colors: intern/extern
3734   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3735   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3736   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3737   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3738   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },

```

```

3739 %----- %-----
3740 linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3741 filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3742 urlbordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3743 menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3744 runbordercolor = [rgb]{0.01961, 0.44314, 0.6902 }
3745 }
3746
3747 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3748 { %all darkgray
3749 linkcolor = [rgb]{0.4 ,0.4 ,0.4 },
3750 filecolor = [rgb]{0.4 ,0.4 ,0.4 },
3751 urlcolor = [rgb]{0.4 ,0.4 ,0.4 },
3752 menucolor = [rgb]{0.4 ,0.4 ,0.4 },
3753 runcolor = [rgb]{0.4 ,0.4 ,0.4 },
3754 %----- %-----
3755 linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3756 filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3757 urlbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3758 menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3759 runbordercolor = [rgb]{0.4 ,0.4 ,0.4 }
3760 }
3761
3762 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3763 { %dvipsnam.def
3764 linkcolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3765 filecolor = [rgb]{1, 0, 0}, %Red
3766 urlcolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3767 menucolor = [rgb]{1, 0, 0}, %Red
3768 runcolor = [rgb]{1, 0, 0}, %Red
3769 %----- %-----
3770 linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3771 filebordercolor = [rgb]{1, 0, 0}, %Red
3772 urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3773 menubordercolor = [rgb]{1, 0, 0}, %Red
3774 runbordercolor = [rgb]{1, 0, 0} %Red
3775 }
3776
3777 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3778 { %dvipsnam.def
3779 linkcolor = [rgb]{0.72, 0, 0}, %BrickRed
3780 filecolor = [rgb]{0, 1, 0}, %Green
3781 urlcolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3782 menucolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3783 runcolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3784 %----- %-----
3785 linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3786 filebordercolor = [rgb]{0, 1, 0}, %Green
3787 urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3788 menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3789 runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3790 }
3791
3792

```

```

3793 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3794 {
3795   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3796   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3797   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3798   menucolor      = [rgb]{0.44706, 0.45882, 0},
3799   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3800 %----- %-----
3801   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3802   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3803   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3804   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3805   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3806 }
3807
3808 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3809 {
3810   linkcolor      = [rgb]{0,0,0},
3811   filecolor      = [rgb]{0,0,0},
3812   urlcolor       = [rgb]{0,0,0},
3813   menucolor      = [rgb]{0,0,0},
3814   runcolor       = [rgb]{0,0,0},
3815 %----- %-----
3816   linkbordercolor = [rgb]{0,0,0},
3817   filebordercolor = [rgb]{0,0,0},
3818   urlbordercolor  = [rgb]{0,0,0},
3819   menubordercolor = [rgb]{0,0,0},
3820   runbordercolor  = [rgb]{0,0,0}
3821 }
3822 </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	
<code>\#</code> .....	280, 791
<code>\\$</code> .....	279
<code>\%</code> .....	792
<code>\-</code> .....	2208
<code>\.</code> .....	551, 555, 557
@curropt commands:	
<code>\@curropt:</code> .....	2729, 2796
<code>\[</code> .....	2208
<code>\]</code> ...	20, 21, 31, 32, 33, 41, 45, 55, 75, 82, 89, 96, 103, 118, 123, 124, 132, 133, 142, 143, 144, 145, 146, 153, 161, 281, 282, 941, 2731, 2798, 3138
<code>\_</code> .....	549, 551, 555, 557
<code>\]</code> .....	2208
A	
<code>\A</code> .....	549, 2208
<code>\Acrobatmenu</code> .....	18, 174
<code>\addcontentsline</code> .....	13
<code>\AddToDocumentProperties</code> ..	389, 2203, 2306
<code>\AddToHook</code> .....	427, 444, 450
<code>\AddToHookNext</code> .....	199
<code>\advance</code> .....	2661, 2736, 2780, 2781
<code>allcolors</code> (hypersetup key) ..	<u>1069</u>
<code>\author</code> .....	2
B	
<code>\b</code> .....	553
<code>\begingroup</code> .....	210, 278, 376, 2567, 2696, 2725, 2737, 2771, 3160, 3591, 3603, 3632, 3662
<code>\belowpdfbookmark</code> .....	3553
<code>\bgroup</code> .....	275, 376
<code>\BOOKMARK</code> .....	3556
<code>bookmarkstype</code> (hypersetup key) ..	13
bool commands:	
<code>\bool_if:NTF</code> .....	268, 297, 356, 709, 735, 754, 774, 785, 822, 887, 952, 1030, 1040, 1293, 1304
<code>\bool_if:nTF</code> .....	902, 1978, 2003, 3297, 3405
<code>\bool_lazy_and:nnTF</code> .....	424
<code>\bool_lazy_or:nnTF</code> .....	1323, 1372
<code>\bool_new:N</code> ...	204, 205, 535, 539, 543
<code>\bool_set_true:N</code> .....	544
<code>bordercolormodel</code> (hypersetup key) ..	13, <u>1090</u>
box commands:	
<code>\box_dp:N</code> .....	646, 2881, 2932, 2967
<code>\box_ht:N</code> .....	645, 2880, 2931, 2966
<code>\box_new:N</code> .....	464, 546
<code>\box_set_dp:Nn</code> .....	2175
<code>\box_set_ht:Nn</code> .....	2182
<code>\box_use:N</code> ...	1310, 2883, 2941, 2970
<code>\box_use_drop:N</code> .....	1315
<code>\box_wd:N</code> .....	644, 2879, 2930, 2965
<code>\l_tmpa_box</code> .....	2877, 2879, 2880, 2881, 2883, 2916, 2930, 2931, 2932, 2941, 2963, 2965, 2966, 2967, 2970
C	
<code>\catcode</code> .....	279, 280
<code>\char</code> .....	2815, 3012, 3024
<code>\chardef</code> .....	159
<code>\cite</code> .....	34
clist commands:	
<code>\clist_item:nn</code> .....	2209, 2238
<code>\clist_map_function:nN</code> .....	125, 134
<code>\clist_map_inline:nn</code> .....	2370
<code>\clist_map_variable:nNn</code> ..	2728, 2795
color commands:	
<code>\color_export:nnN</code> ..	41, 435, 989, 1111
<code>\color_select:n</code> .....	997, 1033, 1314, 2922, 2926
<code>\color_select:nn</code> .....	1003
<code>\color_set:nn</code> .....	2, 394, 1014
<code>\color_set:nnn</code> .....	2, 393, 1020
color names:	
<code>hyp/annot/file</code> .....	<u>532</u>
<code>hyp/annot/link</code> .....	<u>532</u>
<code>hyp/annot/menu</code> .....	<u>532</u>
<code>hyp/annot/run</code> .....	<u>532</u>
<code>hyp/annot/url</code> .....	<u>532</u>
<code>colorfile</code> (hypersetup key) .....	<u>1069</u>
<code>colorlink</code> (hypersetup key) .....	<u>1069</u>
<code>colorlinks</code> (hypersetup key) .....	<u>1046</u>
<code>colormenu</code> (hypersetup key) .....	<u>1069</u>
<code>colorrun</code> (hypersetup key) .....	<u>1069</u>
<code>colorscheme</code> (hypersetup key) .....	1, <u>1481</u>
<code>colorurl</code> (hypersetup key) .....	<u>1069</u>
cs commands:	
<code>\cs_generate_variant:Nn</code> .....	156, 157, 158, 391, 463, 587, 594, 819, 989, 1006, 1023, 2209
<code>\cs_gset:Npn</code> .....	1285
<code>\cs_gset_eq:NN</code> .....	410, 421
<code>\cs_if_exist:NTF</code> .....	13





<code>\l__hyp_annot_Named_bool</code> . . . . .	952	<code>\l__hyp_filename_tmpa_tl</code> . . .	470, 828, 830, 835, 836, 841, 896, 897, 915
<code>\c__hyp_annot_types_seq</code> . . . . .		<code>\__hyp_href_pdf_aux:nn</code> . . . . .	320, 323
. . . . .	477, 533, 537, 1069, 1370, 1468	<code>\l__hyp_href_pdf_destination_tl</code> .	
<code>\l__hyp_annot_URI_bool</code> . . . . .	774	. . . . .	207, 218, 257, 325
<code>\g__hyp_bordercolormodel_str</code> . . .		<code>\l__hyp_href_pdf_page_tl</code> 208, 226, 852	
. . . . .	437, 532, 1094, 1113	<code>\__hyp_href_run_aux:nn</code> . . . . .	337, 340
<code>\__hyp_check_link_nesting:TF</code> . . .		<code>\l__hyp_href_run_parameter_tl</code> . . .	
. . . . .	618, 626, 628, 712, 738, 756, 777, 825, 890, 955	. . . . .	209, 230, 342
<code>\__hyp_citebordercolor_hook_-init:</code>	399, 412, 421	<code>\__hyp_href_url_aux:n</code> . . . . .	363, 366
<code>\__hyp_citecolor_hook_init:</code> . . . . .	397, 401, 410	<code>\__hyp_href_url_aux:nn</code> . . . . .	304, 306
<code>\__hyp_clist_display:n</code> . . . . .	118, 125, 134	<code>\l__hyp_href_url_encode_bool</code> . . .	
<code>\__hyp_color_select:n</code> . . . . .		. . . . .	204, 215, 253, 268, 297, 356
. . . . .	41, 990, 990, 1006	<code>\__hyp_href_url_format:</code> . . . . .	
<code>\__hyp_color_select_aux:wn</code> . . . . .		. . . . .	210, 216, 258, 369
. . . . .	990, 994, 1001	<code>\l__hyp_href_url_ismap_bool</code> . . . . .	
<code>\__hyp_color_set:nn</code> . . . . .	40, 41, 397, 399, 434, 1007, 1007, 1023, 1074, 1110	. . . . .	205, 229, 785
<code>\__hyp_color_set_aux:nwn</code> . . . . .		<code>\l__hyp_href_url_protocol_tl</code> . . .	
. . . . .	1007, 1011, 1018	. . . . .	206, 217, 256, 308, 370
<code>\c__hyp_colorscheme_daleif_prop</code> 3717		<code>\__hyp_if_outer_link:</code> . . . . .	612
<code>\c__hyp_colorscheme_henryford_-prop</code>	3808	<code>\__hyp_if_outer_link:TF</code> . . . . .	628
<code>\c__hyp_colorscheme_julian_prop</code> 3732		<code>\l__hyp_link_Contents_tl</code> . . . . .	
<code>\c__hyp_colorscheme_phelype_prop</code>	3793	. . . . .	27, 35, 36, 512, 520, 527, 719, 790
<code>\c__hyp_colorscheme_szabolcsA_-prop</code>	3762	<code>\__hyp_link_goto_begin:nw</code> . . . . .	
<code>\c__hyp_colorscheme_szabolcsB_-prop</code>	3777	. . . . .	680, 721, 744
<code>\c__hyp_colorscheme_tivv_prop</code> . 3747		<code>\__hyp_link_goto_end:</code> . . . . .	702, 722, 758
<code>\l__hyp_dest_box</code> . . . . .	32, 546, 644, 645, 646, 2159, 2175, 2182	<code>\g__hyp_linknestlevel_int</code> . . . . .	
<code>\l__hyp_dest_name_tmpa_tl</code> . . . . .	470, 683, 684, 690, 694, 696, 699, 845, 858	. . . . .	611, 614, 711, 729, 737, 765, 776, 810, 824, 877, 889, 935, 954, 984
<code>\l__hyp_dest_pdfremotestartview_-tl</code>	496, 853, 1966, 1970	<code>g__hyp_linknestlevel_int</code> . . . . .	611
<code>\g__hyp_dest_pdfstartpage_tl</code> . . .		<code>\c__hyp_map_annot_hyp_prop</code> . . . . .	477
. . . . .	496, 1977, 1979, 1986, 2004, 2011	<code>\c__hyp_map_hyp_annot_prop</code> . . . . .	
<code>\g__hyp_dest_pdfstartview_tl</code> . . .		. . . . .	477, 1024, 1097, 1137, 1164, 1172, 1183, 1210, 1218, 1287, 1395, 1431, 1441, 1554
. . . . .	496, 1979, 1986, 1997, 2001, 2004, 2011	<code>\__hyp_ocg_init:</code> <u>1229</u> , 1229, 1285, 1295	
<code>\l__hyp_dest_pdfview_tl</code> . . . . .		<code>\l__hyp_optlang_regex</code> 2207, 2208, 2240	
. . . . .	531, 662, 671, 2128, 2131, 2138, 2141, 2142, 2143, 2144, 2145, 2146, 2151, 2155, 2191	<code>l__hyp_page/Trans</code> . . . . .	560
<code>\c__hyp_dest_startview_regex</code> . . .		<code>\__hyp_PageLabels_gpush:</code> 595, 595, 608	
. . . . .	547, 1964, 1995	<code>\l__hyp_para_tmpa_tl</code> 473, 901, 904, 915	
<code>\c__hyp_dest_undefined_tl</code> . . . . .		<code>\l__hyp_para_tmpa_tl_uuu\l__hyp_-text_tmpa_str_uuu\g__hyp_text_-tmpa_str</code> . . . . .	470
<code>\__hyp_destination:nn</code> . . . . .		<code>\__hyp_property_record:nn</code> . . . . .	
. . . . .	32, 631, 631, 662, 671	. . . . .	456, 457, 463, 2665
		<code>\__hyp_secondoftwewithopt:wnn</code> . . .	
		. . . . .	382, 383, 384, 385
		<code>\__hyp_setup_info_date_key:nn</code> . . .	
		. . . . .	2292, 2325, 2326
		<code>\__hyp_setup_info_key:nn</code> . . . . .	2210, 2285, 2286, 2287, 2288, 2290, 2291
		<code>\__hyp_store_metadata:nn</code> . . . . .	
		. . . . .	386, 391, 1573, 2217, 2260, 2305, 2320, 2329, 2345, 2413

\_hyp_text_cleanup:N	567, 567, 581	bookmarkstype	13
\l_hyp_text_enc_dest_print_tl	499, 844	bordercolormodel	13, 1090
\l_hyp_text_enc_dest_tl	32, 499, 638, 695	colorfile	1069
\l_hyp_text_enc_file_print_tl	499, 895	colorlink	1069
\l_hyp_text_enc_info_print_tl	499, 519, 526, 592	colorlinks	1046
\l_hyp_text_enc_para_print_tl	499, 900	colormenu	1069
\l_hyp_text_enc_uri_print_tl	270, 273, 299, 302, 358, 361, 499, 782, 1565	colorrund	1069
\_hyp_text_pdfstring:nnN	517, 524, 575, 575, 587, 589, 592, 636, 693, 780, 842, 893, 898, 1565	colorscheme	1, 1481
\_hyp_text_pdfstring_info:nN	590, 590, 594, 2243, 2246, 2266, 2275	colorurl	1069
\_hyp_text_purify:nN	563, 563, 580	debug	1505
\_hyp_text_string_from_unicode:nN	571, 571, 582	destlabel	13
\g_hyp_text_tmpa_str	475, 583, 585	draft	1505
\l_hyp_text_tmpa_str	474, 580, 581, 582, 583	extension	13, 1523
\l_hyp_tmpa_box	464, 1297, 1310, 1315	file	10, 1554
\l_hyp_tmpa_int	464	fileborderstyle	14, 1183
\l_hyp_tmpa_seq	464, 1964, 1966, 1995, 1997, 2118, 2119, 2123, 2125, 2149, 2156, 2157, 2158, 2167, 2169, 2179, 2184, 2240, 2241, 2246, 2616, 2618, 2624	filecolor	1069
\l_hyp_tmpa_str	464, 2243, 2246, 2248, 2250, 2266, 2267, 2269, 2275, 2276, 2280	final	1505
\l_hyp_tmpa_tl	464, 639, 643, 651, 1114, 1118, 1565, 1566, 1571, 1962, 1964, 1993, 1995, 2125, 2126, 2133, 2216, 2217, 2218, 2238, 2243, 2259, 2260, 2266	hidefile	1457
\l_hyp_tmpb_tl	464, 2238, 2240	hidelink	1457
\l_hyp_uri_tmpa_tl	470, 783, 784	hidelinks	1457
hyp/anchor	657	hidemenu	1457
hyp/annot/file (color name)	532	hiderun	1457
hyp/annot/link (color name)	532	hideurl	1457
hyp/annot/menu (color name)	532	hypertexnames	1523
hyp/annot/run (color name)	532	link	10, 1554
hyp/annot/url (color name)	532	linkborder	14
hyp/text/pdfstring	562	linkborderstyle	14, 1183
\hypercalcbp	12, 18, 162	linkcolor	1069
\HyperDestNameFilter	13, 637, 694	linkfileprefix	1523
\hypersetup	1, 2, 5, 9, 10, 13, 19, 68, 101, 181	linktoc	1523
\hypersetup keys:		linktocpage	1523
allcolors	1069	localanchorname	1523
		menu	10, 1554
		menuborder	14
		menuborderstyle	14, 1183
		menucolor	1069
		naturalnames	1523
		nested-links	10
		nesting	14
		ocgcolorfile	1323
		ocgcolorlink	1323
		ocgcolorlinks	1323
		ocgcolormenu	1323
		ocgcolorrund	1323
		ocgcolorurl	1323
		pageanchor	1523
		pdfauthor	2207
		pdfborder	14
		pdfborderstyle	14, 1183
		pdfcreationdate	14, 2292
		pdfcreator	2207
		pdfencoding	1495
		pdfinfo	2359
		pdfkeywords	2207

pdflang	14, 2196	iow commands:
pdflinkmargin	14	\iow_newline: . . . . . 3056, 3062, 3075
pdfmetadate	14, 2292	
pdfmoddate	14, 2292	<b>K</b>
pdfproducer	2207	\kern . . . . . 3578
pdfremotestartview	12	keys commands:
pdfstartview	12	\keys_define:nn 187, 195, 213, 251,
pdfsubject	2207	395, 622, 1046, 1071, 1078, 1090,
pdftitle	2207	1099, 1124, 1139, 1158, 1185, 1204,
pdftrapped	2331	1327, 1341, 1353, 1376, 1382, 1397,
pdfversion	1495	1426, 1457, 1470, 1481, 1495, 1505,
pdfview	12, 2114	1510, 1523, 1535, 1556, 1561, 2114,
plainpages	1523	2196, 2212, 2255, 2294, 2309, 2327,
run	10, 1554	2331, 2359, 2411, 2417, 2430, 2450
runborder	14	\l_keys_key_str . . . . . 192, 2280, 2515
runborderstyle	14, 1183	\keys_set:nn . . . . . 184,
runcolor	1069	221, 240, 267, 296, 319, 336, 355,
unicode	1495	405, 416, 1487, 1494, 2336, 2363,
url	10, 1554	2366, 2367, 2368, 2369, 2440, 2515
urlborder	14	\keys_set_known:nn . . . . . 2518
urlborderstyle	14, 1183	\kvsetkeys . . . . . 183, 2603
urlcolor	1069	
verbose	1505	<b>L</b>
hypertextnames (hypersetup key)	1523	\label . . . . . 9, 13
hypListbox internal commands:		\LayoutCheckField . . . . . 2996
\_hypListbox	2784, 2843	\LayoutChoiceField . . . . . 2766
hypRadio internal commands:		\LayoutPushButtonField . . . . . 2872, 2889
\_hypRadio	2769, 2791	\LayoutTextField . . . . . 2704
hypwritetorep internal commands:		\leavevmode . . . . . 2705, 2802, 3576
\_hypwritetorep	3546	legacy commands:
		\legacy_if:nTF . . . . . 169, 442, 452,
		603, 2641, 2868, 2909, 2959, 2992, 3276
		\let . . . . . 383,
		384, 385, 731, 800, 812, 868, 879,
		928, 937, 986, 2528, 2529, 2556,
		2652, 2653, 2654, 2686, 2687, 2721,
		2722, 2764, 3103, 3110, 3112, 3117,
		3134, 3168, 3587, 3601, 3604, 3613,
		3624, 3631, 3634, 3636, 3644, 3652
		link (hypersetup key) . . . . . 10, 1554
		linkborder (hypersetup key) . . . . . 14
		linkborderstyle (hypersetup key) 14, 1183
		linkcolor (hypersetup key) . . . . . 1069
		linkfileprefix (hypersetup key) . . . 1523
		linktoc (hypersetup key) . . . . . 1523
		linktocpage (hypersetup key) . . . . . 1523
		localanchorname (hypersetup key) . . 1523
		\long . . . . . 7
		<b>M</b>
		\MakeButtonField . . . . .
		. . . . . 2877, 2892, 2916, 2963, 2974
		\MakeChoiceField . . . . . 2852
		\MakeFieldObject . . . . . 2519
<b>I</b>		
\ifcase	3330, 3454, 3489	
\ifcsname	3107	
\ifdim	2734, 2773, 2774	
\ifnum	3161, 3330, 3336, 3367, 3454,	
	3459, 3489, 3494, 3621, 3639, 3654	
\iftrue	3623	
\ifx	2565, 2703, 2763,	
	3138, 3150, 3154, 3171, 3175, 3179,	
	3183, 3187, 3191, 3195, 3199, 3203,	
	3207, 3224, 3228, 3331, 3332, 3340,	
	3344, 3351, 3371, 3375, 3382, 3383,	
	3455, 3463, 3490, 3498, 3571, 3633	
\immediate	2549, 2552	
int commands:		
\int_compare:nNnTF	614, 2123,	
	2149, 2835, 3253, 3306, 3416, 3438	
\int_compare_p:nNn	3299, 3407	
\int_eval:n	851	
\int_gdecr:N	729, 765, 810, 877, 935, 984	
\int_gincr:N	711, 737, 776, 824, 889, 954	
\int_max:nn	852	
\int_new:N	468, 611	

<code>\MakeRadioField</code> .....	2834	<code>\paperwidth</code> .....	3
<code>\MakeTextField</code> .....	2713	<code>\PassOptionsToPackage</code> ...	454, 1515, 1520
<code>\mbox</code> .....	1307	pdf commands:	
menu (hypersetup key) .....	10, 1554	<code>\pdf_bdcobject:nn</code> .....	1309, 1312
menuborder (hypersetup key) .....	14	<code>\pdf_destination:nn</code> ....	27, 156, 650
menuborderstyle (hypersetup key) .....	14, 1183	<code>\pdf_destination:nnnn</code> .....	643
menucolor (hypersetup key) .....	1069	<code>\pdf_emc:</code> .....	1311, 1317
mode commands:		<code>\pdf_link_user:nnn</code> .....	2849
<code>\mode_if_horizontal:TF</code> .....	633, 655	<code>\pdf_name_from_unicode_e:n</code> .....	246, 832, 962
<code>\mode_leave_vertical:</code> ..	264, 293,	<code>\pdf_object_if_exist:nTF</code> .....	830
	316, 333, 352, 682, 794, 860, 917,	<code>\pdf_object_new:n</code> .....	1231,
	960, 2846, 2874, 2891, 2913, 2955, 2998	1232, 1233, 1234, 3043, 3044, 3045	
msg commands:		<code>\pdf_object_ref:n</code> .....	157, 841,
<code>\msg_error:nn</code> .....	426, 2888, 2973	1237, 1239, 1261, 1262, 1265, 1269,	
<code>\msg_info:nnn</code> .....	2224, 2229	1274, 1279, 1284, 2634, 2636, 3101	
<code>\msg_line_context:</code> .....	83	<code>\pdf_object_ref_last:</code> ....	862, 2445
<code>\g_msg_module_name_prop</code> .....	12	<code>\pdf_object_unnamed_write:nn</code> ...	861, 2441
<code>\msg_new:nnn</code> 51, 58, 63, 67, 71, 78,		.....	1235,
85, 92, 99, 106, 112, 119, 128, 138, 149		1241, 1251, 1263, 3046, 3088, 3095	
<code>\msg_new:nnnn</code> .....	15, 26, 37	<code>\pdf_pageobject_ref:n</code> .....	158, 1986, 2011, 2673
<code>\msg_warning:nn</code> .....	171, 1501	<code>\pdf_string_from_unicode:nnN</code> ..	573
<code>\msg_warning:nnn</code> .....	180, 686, 978	<code>\pdf_version:</code> .....	1647, 1766, 1795, 1829, 1844,
<code>\msg_warning:nnnn</code> .....		1870, 1895, 1930, 1946, 2051, 2078	
.....		<code>\pdf_version_compare:NnTF</code> .....	1637, 1728, 1750, 1786, 1819, 1834,
1345, 1386, 1599, 1643,		1860, 1885, 1913, 1936, 2041, 2068	
1677, 1700, 1762, 1791, 1825, 1840,		<code>\pdf_version_compare_p:Nn</code> .....	906, 1324, 1373
1853, 1866, 1891, 1926, 1942, 1969,		.....	168, 1325, 1348, 1374, 1390
2000, 2033, 2047, 2074, 2108, 2190		<code>\pdf_version_minor:</code> ..	167, 1348, 1390
<code>\msg_warning:nnnnn</code> .....	191,	pdfannot commands:	
1417, 1451, 1543, 1619, 1656, 1743,		<code>\pdfannot_box:nnnn</code> .....	2708, 2819, 2878, 2929, 2964, 3028
1776, 1801, 1877, 1904, 1955, 2060,		..	2563, 2579
2087, 2353, 2457, 2476, 2495, 2506		<code>\pdfannot_dict_put:nnn</code> .....	718, 790, 795, 862, 918, 963, 1115,
		1150, 1174, 1196, 1220, 1402, 1433	
		<code>\pdfannot_dict_remove:nn</code> ..	1105,
		1145, 1166, 1191, 1212, 1410, 1443	
		<code>\pdfannot_link:nnn</code> .	796, 863, 919, 964
		<code>\pdfannot_link_goto_begin:nw</code> ..	699
		<code>\pdfannot_link_goto_end:</code> .....	704
		<code>\pdfannot_link_margin:n</code> .....	8, 1681
		<code>\c_pdfannot_link_types_seq</code> ....	541
		pdfauthor (hypersetup key) .....	2207
		<code>\pdfbookmark</code> .....	3555
		pdfborder (hypersetup key) .....	14
		pdfborderstyle (hypersetup key) .	14, 1183
		pdfcreationdate (hypersetup key)	14, 2292
<b>N</b>			
naturalnames (hypersetup key) .....	1523		
nested-links (hypersetup key) .....	10		
nesting (hypersetup key) .....	14		
<code>\newcommand</code> .....	165, 3550, 3552, 3554		
<code>\newcount</code> .....	2657, 2842		
<code>\NewDocumentCommand</code> .....	2519		
<code>\NewExpandableDocumentCommand</code> ....	382		
<code>\newlength</code> .....	3569		
<code>\noexpand</code> ...	2740, 2741, 2742, 2743, 3121		
<code>\nolinkurl</code> .....	4		
<b>O</b>			
ocgcolorfile (hypersetup key) .....	1323		
ocgcolorlink (hypersetup key) .....	1323		
ocgcolorlinks (hypersetup key) ....	1323		
ocgcolormenu (hypersetup key) .....	1323		
ocgcolorrun (hypersetup key) .....	1323		
ocgcolorurl (hypersetup key) .....	1323		
<b>P</b>			
pageanchor (hypersetup key) .....	1523		





<code>\@ifundefined</code>	3126, 3585, 3612, 3622	<code>\Fld@flags</code>	3241,
<code>\@mainaux</code>	2549, 2552, 2572, 2591		3294, 3327, 3363, 3402, 3451, 3486
<code>\@mkboth</code>	3604	<code>\Fld@format@code</code>	3175, 3177
<code>\@ne</code>	2661	<code>\Fld@height</code>	2689, 2710, 2713, 2724, 2782, 2821,
<code>\@nil</code>	3124, 3137, 3143		2834, 2852, 2901, 2949, 2985, 3030
<code>\@part</code>	3631, 3638	<code>\Fld@keystroke@code</code>	3171, 3173
<code>\@pdfauthor</code>	23	<code>\Fld@listcount</code>	2792, 2800, 2835, 2842
<code>\@pdfborder</code>	163	<code>\Fld@mappingname</code>	3154, 3156, 3228, 3230
<code>\@pdfborderstyle</code>	164	<code>\Fld@maxlen</code>	3438, 3440
<code>\@savsf</code>	633, 655	<code>\Fld@menulength</code>	2726, 2732, 2779
<code>\@schapter</code>	3601, 3602	<code>\Fld@name</code>	2685, 2720, 2858, 2981, 3149, 3223
<code>\@secondoftwo</code>	3657	<code>\Fld@onblur@code</code>	3191, 3193
<code>\@sect</code>	3652, 3653	<code>\Fld@onclick@code</code>	3316
<code>\@spart</code>	3644, 3645	<code>\Fld@onenter@code</code>	3203, 3205
<code>\@ssect</code>	3587, 3588	<code>\Fld@onexit@code</code>	3207, 3209
<code>\@tempdima</code>	2727, 2734, 2736, 2773, 2774, 2775, 2779, 2780, 2781, 2782	<code>\Fld@onfocus@code</code>	3187, 3189
<code>\@tempdimb</code>	2733, 2734	<code>\Fld@onmousedown@code</code>	3195, 3197
<code>\@typeset@protect</code>	3571	<code>\Fld@onmouseup@code</code>	3199, 3201
<code>\BKM@color</code>	432, 438	<code>\Fld@pageobjref</code>	2654, 2669, 3106, 3238, 3291, 3324, 3360, 3399, 3448, 3483
<code>\c@secnumdepth</code>	3621, 3639, 3654	<code>\Fld@radiosymbol</code>	3379
<code>\calc@bm@number</code>	3563	<code>\Fld@rotation</code>	3253, 3255, 3299, 3306, 3308, 3330, 3336, 3338, 3367, 3369, 3407, 3416, 3418, 3454, 3459, 3461, 3489, 3494, 3496
<code>\check@bm@number</code>	3562	<code>\Fld@submitflags</code>	3475
<code>\define@key</code>	429	<code>\Fld@validate@code</code>	3179, 3181
<code>\Fld@additionalactions</code>	3213, 3283, 3317, 3353, 3390, 3437, 3477, 3506	<code>\Fld@value</code>	2687, 2703, 2722, 2763, 2764, 3436
<code>\Fld@align</code>	3242, 3328, 3403	<code>\Fld@width</code>	2688, 2702, 2709, 2713, 2723, 2762, 2773, 2775, 2820, 2834, 2852, 2870, 2900, 2911, 2948, 2960, 2984, 2994, 3029
<code>\Fld@altname</code>	3150, 3152, 3224, 3226	<code>\Fld@X@additionalactions</code>	3170, 3215, 3218
<code>\Fld@annotflags</code>	3237, 3290, 3323, 3359, 3398, 3447, 3482	<code>\Form@action</code>	3473
<code>\Fld@annotnames</code>	3222, 3239, 3292, 3325, 3361, 3400, 3449, 3484	<code>\H@old@part</code>	3631, 3642
<code>\Fld@bcolor</code>	3261, 3263, 3344, 3346, 3375, 3377, 3411, 3424, 3426	<code>\H@old@schapter</code>	3601, 3610
<code>\fld@bcolor</code>	3332	<code>\H@old@sect</code>	3652, 3663, 3672
<code>\Fld@bordercolor</code>	3257, 3259, 3301, 3310, 3312, 3331, 3340, 3342, 3371, 3373, 3409, 3420, 3422, 3455, 3463, 3465, 3490, 3498, 3500	<code>\H@old@spart</code>	3644, 3650
<code>\Fld@borderstyle</code>	3243, 3296, 3329, 3365, 3404, 3453, 3504	<code>\H@old@ssect</code>	3587, 3592
<code>\Fld@borderwidth</code>	2780, 2781, 3243, 3296, 3329, 3365, 3404, 3453, 3504	<code>\href@</code>	275, 281
<code>\Fld@calculate@code</code>	2565, 3183, 3185	<code>\href@split</code>	281, 282
<code>\Fld@calculate@sortkey</code>	2576	<code>\Hy@abspage</code>	607
<code>\Fld@cbsymbol</code>	3265	<code>\Hy@activeanchorfalse</code>	678
<code>\Fld@charsize</code>	2779, 3269, 3350, 3381, 3432, 3488	<code>\Hy@activeanchortrue</code>	669
<code>\Fld@checkequals</code>	2731, 2798	<code>\Hy@AtBeginDocument</code>	2530, 2546
<code>\Fld@choices</code>	3352	<code>\Hy@bookmarkstyle</code>	1583
<code>\Fld@color</code>	3270, 3272, 3351, 3382, 3433	<code>\Hy@chapapp</code>	3605, 3616
<code>\Fld@default</code>	2686, 2703, 2721, 2764, 2793, 2794, 2982, 3383, 3387, 3388, 3435	<code>\Hy@colorlink</code>	35
		<code>\Hy@currentbookmarklevel</code>	3544

\Hy@DisableOption	173	\HyAnn@Count	2657, 2658, 2661, 2662, 2663, 2664, 2665, 2666, 2671, 2675
\Hy@drafttrue	1514	\HyField@AddToFields	2559, 2714, 2836, 2853, 2884, 2940, 2969, 3033
\Hy@escapeform	2707, 2804, 2848, 2876, 2915, 2962, 3000, 3108, 3115, 3118	\HyField@afields	2528
\Hy@escapestring	3110, 3112, 3117, 3120, 3121, 3124, 3125, 3134, 3137, 3173, 3177, 3181, 3185, 3189, 3193, 3197, 3201, 3205, 3209, 3265, 3316, 3379, 3435, 3436, 3473	\HyField@AfterAuxOpen	2530, 2556, 2589
\Hy@finaltrue	1519	\HyField@AuxAddToCoFields	2541, 2553, 2574
\Hy@FormObjects	2606, 3041, 3103	\HyField@AuxAddToFields	2534, 2550, 2592
\Hy@gtemp	3166, 3168	\HyField@cofields	2529
\Hy@href	275	\HyField@FlagsCheckBox	2991
\Hy@href@nextactionraw	239	\HyField@FlagsChoice	2772
\Hy@href@page	227	\HyField@FlagsPushButton	2867, 2907, 2958
\Hy@linkfileprefix	1528	\HyField@FlagsRadioButton	2768
\Hy@linktoc	1539	\HyField@FlagsSubmit	2908
\Hy@MakeCurrentHref	200	\HyField@FlagsText	2701
\Hy@MakeCurrentHrefAuto	3589, 3605, 3616, 3646, 3660	\HyField@PDFChoices	2845
\Hy@next	3615, 3624, 3627	\HyField@SetKeys	2697, 2738, 2741, 2760, 2860, 2903, 2951, 2986
\Hy@numberline	166, 3545	\HyPat@ObjRef	3517, 3524, 3531, 3538
\Hy@org@chapter	3613, 3628	\hyper@@link	283
\Hy@OutlineName	3561	\hyper@anchor	657
\Hy@OutlineRerunCheck	3559	\hyper@anchorend	657, 3580, 3607, 3618, 3648
\Hy@pdfmajorversion	168	\hyper@anchorstart	657, 3580, 3607, 3618, 3648
\Hy@pdfminorversion	167	\hyper@link	34, 35, 707
\Hy@pdfstringtrue	30, 578	\hyper@linkend	34, 35, 752
\Hy@pdfversion	3161	\hyper@linkfile	325, 820
\Hy@pstringdef	588, 3141	\hyper@linklaunch	38, 342, 885, 944
\Hy@PutCatalog	595	\hyper@linknamed	40, 176, 950
\Hy@raisedlink	3579, 3606, 3617, 3647	\hyper@linkstart	34, 35, 733
\Hy@RestoreLastskip	654	\hyper@linkurl	308, 369, 772
\Hy@ReturnAfterFi	7, 3142	\hyper@normalise	275, 304, 320, 337, 363, 376
\Hy@safe@activestru	793, 2568	\HyPL@Labels	597, 607
\Hy@SaveLastskip	634	\HyPL@storePageLabel	595
\Hy@secnum@part	3634, 3636, 3639	\HyPsd@SanitizeForOutFile	3548
\Hy@SectionAnchorHref	3570, 3594, 3665	\if@filesw	2548, 2571, 2590
\Hy@SectionHShift	3569, 3578, 3590, 3661	\ifFld@combo	2747, 2777
\Hy@StepCount	2732, 2800	\ifFld@hidden	2702, 2762
\Hy@temp	2569, 2584, 3123, 3124	\ifFld@multiline	2690
\Hy@unicodefalse	3163	\ifFld@popdown	2748
\Hy@VerboseAnchor	635	\ifFld@radio	2744, 2767
\Hy@VerboseLinkStart	714, 740	\ifHy@implicit	3565
\Hy@VerboseLinkStop	724, 760, 803, 870, 930, 974	\ifHy@pdfescapeform	3109, 3119
\Hy@VersionChecked	159	\ifHy@unicode	3162
\Hy@WrapperDef	631	\kv@set@family@handler	179
\Hy@xspace@end	721, 802, 869, 929, 973	\m@ne	3621, 3636
\HyAnn@AbsPageLabel	2653, 2659, 2706, 2803, 2847, 2875, 2914, 2961, 2999	\OBJ@OCG@view	165
		\p@	2736
		\pdf@ifdraftmode	2604

