

# The `hyperref-generic` module

## A generic driver for `hyperref`

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

Version 0.96r, released 2025-05-15

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
urlencode	<code>\hrefurl</code>	if set the code will convert the argument to percent encoding. This allows non-ascii input.
protocol	<code>\hrefurl</code> , <code>\url</code>	This sets a prefix/protocol that is added to the url, see below.
format	<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.
destination	<code>\href</code> , <code>\hrefpdf</code>	A destination name in the PDF
page	<code>\href</code> , <code>\hrefpdf</code>	destination page, default: 1
pdfremotestartview	<code>\href</code> , <code>\hrefpdf</code>	start view, default: Fit
ismap	<code>\href</code> , <code>\hrefurl</code>	see PDF reference
afrelationship	<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.
run-parameter	<code>\hreflaunch</code>	run parameter (see the PDF reference)
nextactionraw	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

---

```

\l_hyp_annot_colorlink_bool
\l_hyp_annot_colorurl_bool
\l_hyp_annot_colorfile_bool
\l_hyp_annot_colorrerun_bool
\l_hyp_annot_colormenu_bool
\l_hyp_annot_ocgcolorlink_bool
\l_hyp_annot_ocgcolorurl_bool
\l_hyp_annot_ocgcolorfile_bool
\l_hyp_annot_ocgcolorrerun_bool
\l_hyp_annot_ocgcolormenu_bool

```

---

These boolean are used by the `colorlinks` and `ocgcolorlinks` and related keys. These keys insert 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 new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference without link (this is naturally also possible with `\ref*{abc}`). This disables also all hooks of the link type, so the link is for example no longer colored. It also removes the implicit grouping of the content.

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

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

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

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

- `pdfview` will use the values to calculate coordinates relative to the current location. So `0 -100 200 400` will give a “box” of width 200bp, height 400bp and depth 100dp that the destination should encompass. Missing numbers will be set to 0. But one should be aware that is 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 dvi<sub>pd</sub>mx 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}[2025-05-15 v0.96r %
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\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  
`\c__hyp_map_annot_hyp_prop` annot names to hyperref names and back.  
`\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  
`\g__hyp_dest_pdfstartview_tl` the current and remote files. The instruction is in “PDF format” but without the leading  
`\l__hyp_dest_pdfremotestartview_tl` slash!

```
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 put something into the relevant annotation dictionaries.

```

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:nn {hyp/link/GoToR/Contents}{1}
516 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
517 {
518   \__hyp_text_pdfstring:eoN
519   { Go-to-destination~#1 }
520   { \l__hyp_text_enc_info_print_tl }
521   \l__hyp_link_Contents_tl
522   \pdfannot_dict_put:nne {link/GoTo}{Contents}
523   {\l__hyp_link_Contents_tl}
524 }
525 \socket_new_plug:nnn {hyp/link/GoToR/Contents}{default}
526 {
527   \__hyp_text_pdfstring:eoN
528   { Open~file~#1 }
529   { \l__hyp_text_enc_info_print_tl }
530   \l__hyp_link_Contents_tl
531   \pdfannot_dict_put:nne {link/GoToR}{Contents}
532   {\l__hyp_link_Contents_tl}
533 }
534 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
535 {
536   \__hyp_text_pdfstring:eoN
537   { #1 }
538   { \l__hyp_text_enc_info_print_tl }
539   \l__hyp_link_Contents_tl
540   \pdfannot_dict_put:nne {link/URI}{Contents}
541   {\l__hyp_link_Contents_tl}
542 }
543 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
544 \socket_assign_plug:nn{hyp/link/GoToR/Contents}{default}
545 \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.

```

546 \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  
 hyp/annot/url (color name) end when the color scheme is used  
 hyp/annot/file (color name)  
 hyp/annot/run (color name)  
 hyp/annot/menu (color name)

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

```
547 \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.

```
548 \seq_map_inline:Nn \c_hyp_annot_types_seq
549 {
550   \bool_new:c {l_hyp_annot_color#1_bool}
551 }
```

(End of definition for `\l_hyp_annot_colorlink_bool` and others. These variables are documented on page 7.)

`\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.

```
552 \seq_map_inline:Nn \c_hyp_annot_types_seq
553 {
554   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
555 }
```

(End of definition for `\l_hyp_annot_ocgcolorlink_bool` and others. These variables are documented on page 7.)

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

```
556 \seq_map_inline:Nn \c_pdfannot_link_types_seq
557 {
558   \bool_new:c {l_hyp_annot_#1_bool}
559   \bool_set_true:c {l_hyp_annot_#1_bool}
560 }
```

(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.

```
561 \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.

```
562 \regex_const:Nn \c__hyp_dest_startview_regex
563 {
564   \A\ *
565   (?
566     (? :XYZ (? :\ +(?: (? :\d+|\d*\.\d+) |null)) {3}\ )
567     |
568     (? :Fit\b|FitB\b)
569     |
570     (? :(? :FitH|FitV|FitBH|FitBV) (? :\ +(?: \d+|\d*\.\d+) |\ +null) {1})
571     |
572     (? :FitR (? :\ +\d+|\ +\d*\.\d+) {4}\ )
573   )
574 }
```

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

## 7.7 PDF dictionaries

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

```
575 \pdfdict_new:n {l__hyp_page/Trans}
576 \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`

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

*(End of definition for `hyp/text/pdfstring`.)*

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`

```
578 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
579 {
580   \str_set:Ne #2 {\text_purify:n { #1 } }
581 }
```

*(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`

```
582 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
583 {
584
585 }
```

*(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`.

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

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

`\__hyp_text_string_from_unicode:nN`

```
586 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
587 {
588   \pdf_string_from_unicode:nVN { #1 } #2 #2
589 }
```

*(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`

```
590 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
591 {
592   \group_begin:
593   \Hy@pdfstringtrue
594   \hook_use:n {hyp/text/pdfstring}
595   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
596   \__hyp_text_cleanup:N      \l__hyp_text_tmpa_str
597   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
598   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
599   \group_end:
600   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
601 }
602 \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

```
603 \cs_new_protected:Npn \Hy@pstringdef #1 #2
604 { \__hyp_text_pdfstring:enN {#2} {utf8/string-raw}#1 }
```

This is a special version for info keys:

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

```
605 \cs_new_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
606 {
607   \_hyp\_text\_pdfstring:non { #1 }{\ \_hyp\_text\_enc\_info\_print\_tl } #2
608 }
609 \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:** hxdetex.def, line 80-110

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

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

**dvipdfm:** identical to xetex

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

**luatex** identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-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
610 \cs_new_protected:Npn \_hyp\_PageLabels\_gpush:
611 {
612   \pdfmanagement\_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
613 }
614
615 \def\Hy@PutCatalog #1 {}
616
617
618 \legacy\_if:nT { Hy@pdfpagelabels }
```

```

619 {
620   \cs_set_protected:Npn \HyPL@StorePageLabel #1
621     {
622       \tl_gput_right:Nc \HyPL@Labels { \the\Hy@abspage<<#1>> }
623       \__hyp_PageLabels_gpush:
624     }
625 }

```

(End of definition for `\__hyp_PageLabels_gpush:`, `\Hy@PutCatalog`, and `\HyPL@StorePageLabel`.)

## 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`

```

626 \int_new:N \g__hyp_linknestlevel_int

```

(End of definition for `g__hyp_linknestlevel_int`.)

```

627 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
628 {
629   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
630     { \prg_return_false: }
631     { \prg_return_true: }
632 }
633 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
634 {
635   \use_i:nn {#1}{#2}
636 }
637 \keys_define:nn { hyp }
638 {
639   nested-links .choice:,
640   nested-links / true .code:n =
641     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },

```



```

642     nested-links / false .code:n =
643     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
644     nested-links .default:n = {true}
645 }

```

## 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 `\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
646 \Hy@WrapperDef \__hyp_destination:nn #1 #2
647 {
648   \mode_if_horizontal:T { \@savsf\spacefactor }
649   \Hy@SaveLastskip      %defined in hyperref
650   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
651   \__hyp_text_pdfstring:eoN
652   { \HyperDestNameFilter{#1} }
653   { \l__hyp_text_enc_dest_tl }
654   \l__hyp_tmpa_tl
655   \str_if_eq:nnTF {#2} {fitrbx}
656   {
657     \exp_args:NV
658     \pdf_destination:nnnn \l__hyp_tmpa_tl
659     { \box_wd:N \l__hyp_dest_box }
660     { \box_ht:N \l__hyp_dest_box }
661     { \box_dp:N \l__hyp_dest_box }
662   }
663   {
664     \exp_args:NV
665     \pdf_destination:nf
666     { \l__hyp_tmpa_tl }
667     { #2 }
668   }
669   \Hy@RestoreLastskip %defined in hyperref
670   \mode_if_horizontal:T { \spacefactor\@savsf }
671 }

```

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

These 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 672 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 673 \hook_new:n{hyp/anchor}
hyp/anchor 674 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 675 {
676   \exp_args:NnV
677   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
678   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
679   \hook_use:n{hyp/anchor}
680 }
681
682 \cs_new_protected:Npn \hyper@anchorstart #1
683 {
684   \Hy@activeanchortrue
685   \exp_args:NnV
686   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
687   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
688   \hook_use:n{hyp/anchor}
689 }
690
691 \cs_new_protected:Npn \hyper@anchorend
692 {
693   \Hy@activeanchorfalse
694 }

```

(End of definition for `\hyper@anchor` and others.)

### 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:

```

695 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
696 {
697   \mode_leave_vertical:
698   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
699   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
700   {
701     \msg_warning:nne
702     { hyp }
703     { empty-destination-name }
704     { \c__hyp_dest_undefined_tl }
705     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
706   }
707   {
708     \__hyp_text_pdfstring:eoN
709     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
710     { \l__hyp_text_enc_dest_tl }
711     \l__hyp_dest_name_tmpa_tl
712   }
713   \exp_args:No

```

```

714     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
715   }
716
717   \cs_new_protected:Npn \__hyp_link_goto_end:
718     {
719     \pdfannot_link_goto_end:
720   }

```

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

```

721
722   \cs_new_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
723     {
724     \bool_if:NTF \l__hyp_annot_GoTo_bool
725       {
726       \int_gincr:N\g__hyp_linknestlevel_int
727       \__hyp_check_link_nesting:TF
728         {
729         \Hy@VerboseLinkStart{#1}{#2}
730         \group_begin:
731         \tl_set:Nn \l_hyp_current_dest_name_tl {#2}

```

this socket adds something to the `/Contents` key.

```

732         \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{#3}
733         \hook_use:n {hyp/link/#1}
734         \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
735         \__hyp_link_goto_end:
736         \group_end:
737         \Hy@VerboseLinkStop
738       }
739     {
740     \group_begin: #3\group_end:

```

```

741     }
742     \int_gdecr:N\g__hyp_linknestlevel_int
743   }
744   {\let\protect\relax#3}}
745 }
746 \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
747 {
748   \bool_if:NT \l__hyp_annot_GoTo_bool
749   {
750     \int_gincr:N\g__hyp_linknestlevel_int
751     \__hyp_check_link_nesting:TF
752     {
753       \Hy@VerboseLinkStart{#1}{#2}% only for debug
754       \group_begin:
755       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
756       \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{}
757       \hook_use:n {hyp/link/#1}
758       \__hyp_link_goto_begin:nw {#2}
759     }
760     {
761       \group_begin:
762     }
763   }
764 }
765
766 \cs_new_protected:Npn \hyper@linkend
767 {
768   \bool_if:NT \l__hyp_annot_GoTo_bool
769   {
770     \__hyp_check_link_nesting:TF
771     {
772       \__hyp_link_goto_end:
773       \group_end:
774       \Hy@VerboseLinkStop
775     }
776     {
777       \group_end:
778     }
779     \int_gdecr:N\g__hyp_linknestlevel_int
780   }
781 }

```

## 10.4 URI links

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

```

782 \pdfdict_new:n {l_hyp/annot/A/URI}
783 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
784 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
785
786 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
787 {
788   \bool_if:NTF \l__hyp_annot_URI_bool
789   {
790     \int_gincr:N\g__hyp_linknestlevel_int

```

```

791     \__hyp_check_link_nesting:TF
792     {
793         \group_begin:
794         \__hyp_text_pdfstring:eoN
795         { #2}
796         { \l__hyp_text_enc_uri_print_tl }
797         \l__hyp_uri_tmpa_tl
798         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
799         \bool_if:NT \l__hyp_href_url_ismap_bool
800         {
801             \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
802         }

```

This socket adds something to the /Contents key.

```

803     \socket_use:nn{hyp/link/URI/Contents}{#2}
804     \cs_set_eq:NN \# \c_hash_str
805     \cs_set_eq:NN \% \c_percent_str
806     \Hy@safe@activestrue
807     \mode_leave_vertical:
808     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
809     \pdfannot_link:nen { URI }
810     {
811     }
812     {
813         \let\protect\relax
814         #1
815         \Hy@xspace@end
816         \Hy@VerboseLinkStop %where is the start??
817     }
818     \group_end:
819     }
820     {
821         \group_begin: #1 \group_end:
822     }
823     \int_gdecr:N\g__hyp_linknestlevel_int
824     }
825     {\let\protect\relax#1}}
826 }
827

```

## 10.5 GoToR Links files

```

828 \pdfdict_new:n {l_hyp/annot/A/GoToR}
829 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
830 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
831
832 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
833 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
834 {
835     \bool_if:NTF \l__hyp_annot_GoToR_bool
836     {
837         \int_gincr:N\g__hyp_linknestlevel_int
838         \__hyp_check_link_nesting:TF
839         {

```

```

840 \group_begin:
841 \tl_set:Nc \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
842 \exp_args:Nc
843 \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
844 {
845 \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n
846 \pdffile_embed_file:noe
847 {}
848 {\l__hyp_filename_tmpa_tl }
849 {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
850 }
851 \pdfdict_put:nne
852 {l_hyp/annot/A/GoToR}
853 {F}
854 {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
855 \__hyp_text_pdfstring:nnN
856 { #3 }
857 { \l__hyp_text_enc_dest_print_tl }
858 \l__hyp_dest_name_tmpa_tl

```

This socket adds something to the /Contents key.

```

859 \socket_use:nn{hyp/link/GoToR/Contents}{#2}
860 \tl_if_blank:eTF {#3}
861 {
862 \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
863 {
864 [
865 \int_eval:n
866 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
867 /\l__hyp_dest_pdfremotestartview_tl
868 ]
869 }
870 }
871 {
872 \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
873 }
874 \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>

```

875 \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
876 \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
877 \pdfannot_link:nnn %expansion??
878 { GoToR }
879 {
880 }
881 {
882 \let\protect\relax
883 #1\Hy@xspace@end
884 \Hy@VerboseLinkStop %where is the start??
885 }
886 \group_end:
887 }
888 {
889 \group_begin: #1 \group_end:

```

```

890     }
891     \int_gdecr:N\g__hyp_linknestlevel_int
892   }
893   {{\let\protect\relax#1}}
894 }

```

## 10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

895 \pdfdict_new:n {l_hyp/annot/A/Launch}
896 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
897 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
898
899 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
900 {
901   \bool_if:NTF \l__hyp_annot_Launch_bool
902   {
903     \int_gincr:N\g__hyp_linknestlevel_int
904     \__hyp_check_link_nesting:TF
905     {
906       \group_begin:
907         \__hyp_text_pdfstring:nnN
908         { #1 }
909         { \l__hyp_text_enc_file_print_tl }
910         \l__hyp_filename_tmpa_tl
911         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
912         \__hyp_text_pdfstring:noN
913         { #3 }
914         { \l__hyp_text_enc_para_print_tl }
915         \l__hyp_para_tmpa_tl
916       \bool_if:nTF
917       {
918         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {()}
919         ||
920         \pdf_version_compare_p:Nn > {1.9}
921       }
922       {
923         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
924       }
925       {
926         \pdfdict_put:mne
927         {l_hyp/annot/A/Launch}
928         {Win}
929         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
930       }
931     \mode_leave_vertical:
932     \pdfannot_dict_put:mne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
933     \pdfannot_link:nen
934     { Launch }
935     {
936       % /A
937       % <<
938       % \pdfdict_use:n {l_hyp/annot/A/Launch}
939       % >>
940     }

```

```

941         {
942             \let\protect\relax
943             #2\Hy@xspace@end
944             \Hy@VerboseLinkStop %where is the start??
945         }
946     \group_end:
947 }
948 { \group_begin: #2 \group_end: }
949 \int_gdecr:N\g__hyp_linknestlevel_int
950 }
951 {\let\protect\relax#2}}
952 }

```

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

```

953 \use:e
954 { % filename, anchor text, linkname
955   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
956 }
957 {
958   \hyper@linklaunch {#1}{#2}{#3}
959 }

```

## 10.7 Named links (menu)

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

```

960 \pdfdict_new:n {l_hyp/annot/A/Named}
961 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
962 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
963
964 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
965 {
966   \bool_if:NTF \l__hyp_annot_Named_bool
967   {
968     \int_gincr:N\g__hyp_linknestlevel_int
969     \__hyp_check_link_nesting:TF
970     {
971       \group_begin:
972       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
973       {
974         \mode_leave_vertical:
975         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
976           {\pdf_name_from_unicode_e:n{#1}}
977         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
978         \pdfannot_link:nnn { Named }
979         {
980           %           /A
981           %           <<
982           %           \pdfdict_use:n { l_hyp/annot/A/Named }
983           %           >>
984         }
985       }
986       #2
987       \Hy@xspace@end

```



```

988         \Hy@VerboseLinkStop
989     }
990 }
991 {
992     \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
993     #2
994 }
995 \group_end:
996 }
997 { \group_begin: #2 \group_end: }
998 \int_gdecr:N\g__hyp_linknestlevel_int
999 }
1000 {\let\protect\relax#2}}
1001 }
1002

```

## 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.

```

1003 \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.

```

```

1004 \cs_new_protected:Npn \__hyp_color_select:n #1
1005 {
1006     \tl_if_head_eq_charcode:nNTF {#1}[ %]
1007     {
1008         \__hyp_color_select_aux:wn #1
1009     }
1010     {
1011         \color_select:n {#1}
1012     }
1013 }
1014
1015 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
1016 {
1017     \color_select:nn {#1}{#2}
1018 }
1019
1020 \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` `\_hyp_color_set_aux:nwn` Color keys need to parse color expressions. Two input types are supported: `color=[rgb]{1,0,.5}` and `color=red!50!blue`.

```

1021 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
1022 {
1023   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1024   {
1025     \_hyp_color_set_aux:nwn { #1 } #2
1026   }
1027   {
1028     \color_set:nn {#1} {#2}
1029   }
1030 }
1031
1032 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
1033 {
1034   \color_set:nnn {#1}{#2}{#3}
1035 }
1036
1037 \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.

```

1038 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1039 {
1040   \hook_gput_code:nnn
1041     {pdfannot/link/#2/begin}
1042     {hyp/color}
1043     {
1044       \bool_if:cT { l_hyp_annot_color#1_bool }
1045       {
1046         \group_begin:
1047         \color_select:n { hyp/color/#1}
1048       }
1049     }
1050   \hook_gput_code:nnn
1051     {pdfannot/link/#2/end}
1052     {hyp/color}
1053     {
1054       \bool_if:cT { l_hyp_annot_color#1_bool }
1055       {
1056         \group_end:
1057       }
1058     }

```

```

1058     }
1059 }

```

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

```

1060 \keys_define:nn { hyp }
1061 {
1062   ,colorlinks .choice:
1063   ,colorlinks / true .meta:n =
1064   {
1065     ,pdfborder={0~0~0}
1066     ,pdfborderstyle=
1067     ,colorurl =#1
1068     ,colorlink =#1
1069     ,colorryn =#1
1070     ,colormenu =#1
1071     ,colorfile =#1
1072   }
1073   ,colorlinks / false .meta:n =
1074   {
1075     ,colorurl =#1
1076     ,colorlink =#1
1077     ,colorryn =#1
1078     ,colormenu =#1
1079     ,colorfile =#1
1080   }
1081   ,colorlinks .default:n = {true}
1082 }

```

```

colorurl (setup key)
colorlink (setup key) 1083 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorryn (setup key) 1084 {
colormenu (setup key) 1085   \keys_define:nn { hyp }
colorfile (setup key) 1086   {
  urlcolor (setup key) 1087     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1088     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
  runcolor (setup key) 1089   }
  menucolor (setup key) 1090 }
  filecolor (setup key) 1091
allcolors (setup key) 1092 \keys_define:nn { hyp }
  {
1093   ,allcolors .meta:n =
1094   {
1095     ,urlcolor=#1
1096     ,linkcolor=#1
1097     ,runcolor=#1
1098     ,filecolor=#1
1099     ,menucolor=#1
1100   }
1101   ,allcolors .value_required:n = true
1102 }
1103 }

```

## 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*)

```
1104 \keys_define:nn { hyp }
1105   {
1106     ,bordercolormodel .choices:nn =
1107       {rgb,cmyk}
1108       { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1109     ,bordercolormodel .initial:n ={rgb}
1110   }

1111 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1112   {
1113     \keys_define:nn { hyp }
1114     {
1115       #1bordercolor .code:n =
1116       {
1117         \tl_if_empty:nTF { ##1 }
1118         {
1119           \pdfannot_dict_remove:nn
1120             {link/#2}
1121             { C }
1122         }
1123         {
1124           \__hyp_color_set:ne {hyp/color/#1border}{##1}
1125           \color_export:nVN
1126             {hyp/color/#1border}
1127           \g__hyp_bordercolormodel_str
1128           \l__hyp_tmpa_tl
1129           \pdfannot_dict_put:nne
1130             {link/#2}
1131             { C }
1132             { [\l__hyp_tmpa_tl] }
1133         }
1134       }
1135     }
1136   }

1137
1138 \keys_define:nn { hyp }
1139   {
1140     ,allbordercolors .meta:n =
1141     {
1142       ,linkbordercolor=#1
1143       ,urlbordercolor =#1
1144       ,filebordercolor=#1
1145       ,menubordercolor=#1
1146       ,runbordercolor =#1
1147     }
1148     ,allbordercolors .value_required:n = true
1149   }
```

1150

### 11.3.2 Borderwidth and -arc

```
1151 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1152 {
1153   \keys_define:nn { hyp }
1154   {
1155     #1border .code:n =
1156     {
1157       \tl_if_empty:nTF { ##1 }
1158       {
1159         \pdfannot_dict_remove:nn
1160         {link/#2}
1161         { Border }
1162       }
1163       {
1164         \pdfannot_dict_put:nnn
1165         {link/#2}
1166         { Border }
1167         { [##1] }
1168       }
1169     }
1170   }
1171 }
1172 \keys_define:nn { hyp }
1173 {
1174   ,pdfborder .code:n =
1175   {
1176     \tl_if_empty:nTF { #1 }
1177     {
1178       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1179       {
1180         \pdfannot_dict_remove:nn
1181         {link/##2}
1182         { Border }
1183       }
1184     }
1185     {
1186       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1187       {
1188         \pdfannot_dict_put:nnn
1189         {link/##2}
1190         { Border }
1191         { [#1] }
1192       }
1193     }
1194   }
1195   ,pdfborder .initial:n = {0~0~1},
1196 }
```

### 11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1197 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1198 {
runborderstyle (setup key) 1199   \keys_define:nn { hyp }
fileborderstyle (setup key) 1200   {
menuborderstyle (setup key) 1201     #1borderstyle .code:n =
1202     {
1203       \tl_if_empty:nTF { ##1 }
1204       {
1205         \pdfannot_dict_remove:nn
1206         {link/#2}
1207         { BS }
1208       }
1209       {
1210         \pdfannot_dict_put:nnn
1211         {link/#2}
1212         { BS }
1213         { <<##1>> }
1214       }
1215     }
1216   }
1217 }
1218 \keys_define:nn { hyp }
1219 {
1220   ,pdfborderstyle .code:n =
1221   {
1222     \tl_if_empty:nTF { #1 }
1223     {
1224       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1225       {
1226         \pdfannot_dict_remove:nn
1227         {link/##2}
1228         { BS }
1229       }
1230     }
1231     {
1232       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1233       {
1234         \pdfannot_dict_put:nnn
1235         {link/##2}
1236         { BS }
1237         { <<##1>> }
1238       }
1239     }
1240   }
1241   ,pdfborderstyle .initial:n = {},
1242 }

```

## 11.4 ocgcolorlinks

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

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

```

1243 \cs_new_protected:Npn \__hyp_ocg_init:
1244 {
1245   \pdf_object_new:n { __hyp/OCG/View }
1246   \pdf_object_new:n { __hyp/OCG/Print }
1247   \pdf_object_new:n { __hyp/OCG/config }
1248   \pdf_object_new:n { __hyp/OCG/refarray }
1249   \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1250   {
1251     \pdf_object_ref:n { __hyp/OCG/View }
1252     \c_space_tl
1253     \pdf_object_ref:n { __hyp/OCG/Print }
1254   }
1255   \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1256   {
1257     /Type/OCG
1258     /Name(View)
1259     /Usage
1260     <<
1261       /Print <</PrintState/OFF>>~
1262       /View <</ViewState/ON >>~
1263     >>
1264   }
1265   \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1266   {
1267     /Type/OCG
1268     /Name(Print)
1269     /Usage
1270     <<
1271       /Print <</PrintState/ON>>~
1272       /View <</ViewState/OFF>>~
1273     >>
1274   }
1275   \pdfmanagement_add:nne { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View} }
1276   \pdfmanagement_add:nne { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1277   \pdf_object_write:nne { __hyp/OCG/config } { dict }
1278   {
1279     /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1280     /AS[
1281       <<
1282         /Event/View
1283         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1284         /Category[/View]
1285       >>
1286       <<
1287         /Event/Print
1288         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1289         /Category[/Print]
1290       >>
1291       <<
1292         /Event/Export
1293         /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1294         /Category[/Print]

```

```

1295         >>
1296     ]
1297 }
1298 \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1299 \cs_gset:Npn \__hyp_ocg_init: {}
1300 }

```

(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.

```

1301 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1302 {
1303     \hook_gput_code:nnn
1304     {pdfannot/link/#2/begin}
1305     {hyp/ocg}
1306     {
1307         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1308         {
1309             \__hyp_ocg_init:
1310             \group_begin:
1311             \hbox_set:Nw \l__hyp_tmpa_box
1312         }
1313     }
1314     \hook_gput_code:nnn
1315     {pdfannot/link/#2/end}
1316     {hyp/ocg}
1317     {
1318         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1319         {
1320             \hbox_set_end:
1321             \mbox
1322             {
1323                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1324                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1325                 \pdf_emc:
1326                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1327                 \group_begin:
1328                 \color_select:n { hyp/color/#1 }
1329                 \box_use_drop:N \l__hyp_tmpa_box
1330                 \group_end:
1331                 \pdf_emc:
1332             }
1333             \group_end:
1334         }
1335     }
1336 }

```

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

```

ocgcolorlink (setup key) 1337 \bool_lazy_or:nnTF
  ocgcolorurl (setup key) 1338 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1339 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1340 {
  ocgcolorrun (setup key) 1341     \keys_define:nn { hyp }
1342     {

```



```

1343     ,_ocgcolorlinks .meta:n =
1344     {
1345         ocgcolorlink=#1,
1346         ocgcolorurl=#1,
1347         ocgcolorfile=#1,
1348         ocgcolorruncolor=#1,
1349         ocgcolormenu=#1
1350     }
1351     ,_ocgcolorlinks .default:n = true
1352 }
1353 }
1354 {
1355     \keys_define:nn { hyp }
1356     {
1357         ,_ocgcolorlinks .code:n =
1358         {
1359             \msg_warning:nnee
1360             { hyp }
1361             { ignore-deprecated-or-unknown-option-in-pdf-version }
1362             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1363         }
1364     }
1365 }
1366
1367 \keys_define:nn { hyp }
1368 {
1369     ,ocgcolorlinks .choice:
1370     ,ocgcolorlinks / true .meta:n =
1371     {
1372         pdfborder      ={{0-0-0}},
1373         pdfborderstyle ={{}},
1374         colorlinks     = false,
1375         _ocgcolorlinks = true
1376     }
1377     ,ocgcolorlinks / false .meta:n =
1378     {
1379         _ocgcolorlinks = false
1380     }
1381     ,ocgcolorlinks .default:n = {true}
1382 }
1383
1384 \seq_map_inline:Nn \c__hyp_annot_types_seq
1385 {
1386     \bool_lazy_or:nnTF
1387     { \pdf_version_compare_p:Nn > {1.4} }
1388     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1389     {
1390         \keys_define:nn { hyp }
1391         {
1392             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1393         }
1394     }
1395     {
1396         \keys_define:nn { hyp }

```

```

1397     {
1398         ,ocgcolor#1 .code:n=
1399         {
1400             \msg_warning:nnee
1401             { hyp }
1402             { ignore-deprecated-or-unknown-option-in-pdf-version }
1403             { ocgcolor#1 }
1404             { \pdf_version_major:\pdf_version_minor: }
1405         }
1406     }
1407 }
1408 }

```

## 11.5 Highlighting

This keys set what happens if you click on a link

```

1409 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1410 {
1411     \keys_define:nn { hyp }
1412     {
1413         ,#1highlight .choices:nn =
1414         { /I, /N, /O, /P}
1415         {
1416             \pdfannot_dict_put:nnn
1417             {link/#2}
1418             { H }
1419             { ##1 }
1420         }
1421     }
1422     ,#1highlight / .code:n =
1423     {
1424         \pdfannot_dict_remove:nn
1425         {link/#2}
1426         { H }
1427     }
1428     ,#1highlight / unknown .code:n =
1429     {
1430         \msg_warning:nnee { hyp } { unknown-choice+empty }
1431         { #1highlight }
1432         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1433         { \exp_not:n {##1} }
1434     }
1435 }
1436 }
1437 }
1438
1439
1440 \keys_define:nn { hyp }
1441 {
1442     ,pdfhighlight .choices:nn =
1443     { /I, /N, /O, /P}
1444     {
1445         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1446         {

```

```

1447         \pdfannot_dict_put:nnn
1448         {link/###2}
1449         { H }
1450         { #1 }
1451     }
1452 }
1453 ,pdfhighlight / .code:n =
1454 {
1455     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1456     {
1457         \pdfannot_dict_remove:nn
1458         {link/##2}
1459         { H }
1460     }
1461 }
1462 ,pdfhighlight .initial:n = {/I},
1463 ,pdfhighlight / unknown .code:n =
1464 {
1465     \msg_warning:nnee { hyp } { unknown-choice+empty }
1466     { pdfhighlight }
1467     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1468     { \exp_not:n {#1} }
1469 }
1470 }

```

## 11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1471 \keys_define:nn { hyp }
hideurl (setup key) 1472 {
hidefile (setup key) 1473     hidelinks .meta:n =
hiderun (setup key) 1474     {
hidemenu (setup key) 1475         ,colorlinks      = false
1476         ,ocgcolorlinks = false
1477         ,pdfborder      = { 0~0~0 }
1478         ,pdfborderstyle=
1479     }
1480 }
1481
1482 \seq_map_inline:Nn \c__hyp_annot_types_seq
1483 {
1484     \keys_define:nn { hyp }
1485     {
1486         hide#1 .meta:n =
1487         {
1488             ,color#1      = false
1489             ,ocgcolor#1  = false
1490             ,#1border     = { 0~0~0 }
1491             ,#1borderstyle =
1492         }
1493     }
1494 }

```

## 11.7 color schemes and settings

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

`colorscheme` (*setup key*)

```
1495 \keys_define:nn { hyp }
1496 {
1497   colorscheme .code:n =
1498   {
1499     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1500     {
1501       \keys_set:nn { hyp }
1502       {
1503         ##1 = ##2
1504       }
1505     }
1506   }
1507 }
1508 \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*)

```
1509 \keys_define:nn { hyp }
1510 {
1511   ,unicode .code:n = {}
1512   ,pdfencoding .code:n = {}
1513   ,pdfversion .code:n =
1514   {
1515     \msg_warning:nn { hyp }{ pdfversion-disabled }
1516   }
1517 }
1518 %
```

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

This keys are typically set only once.

`verbose` (*setup key*)

`debug` (*setup key*)

`draft` (*setup key*)

`final` (*setup key*)

```
1519 \keys_define:nn { hyp }
1520 {
1521   ,verbose .legacy_if_set:n = {Hy@verbose}
1522   ,debug .legacy_if_set:n = {Hy@verbose}
1523 }
1524 \keys_define:nn { hyp }
1525 {
1526   ,draft .code:n =
1527   {
1528     \Hy@drafttrue
```

```

1529         \PassOptionsToPackage{draft}{bookmark}
1530     }
1531     ,final .code:n =
1532     {
1533         \Hy@finaltrue
1534         \PassOptionsToPackage{final}{bookmark}
1535     }
1536 }

extension (setup key)
hypertextnames (setup key) 1537 \keys_define:nn { hyp }
naturalnames (setup key) 1538 {
pageanchor (setup key) 1539     ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1540     ,extension .initial:n= pdf
linktocpage (setup key) 1541     ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1542     ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1543     ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1544     ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1545     ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1546     ,plainpages .legacy_if_set:n = {Hy@plainpages}
1547 }
1548
1549 \keys_define:nn { hyp }
1550 {
1551     ,linktoc .choices:nn = { none, section, all, page }
1552     {
1553         \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1554     }
1555     ,linktoc / unknown .code:n =
1556     {
1557         \msg_warning:nnee { hyp } { unknown-choice }
1558         { linktoc }
1559         { none, section, all, page }
1560         { \exp_not:n {#1} }
1561     }
1562     ,linktocpage .choice:
1563     ,linktocpage / true .meta:n = {linktoc=page}
1564     ,linktocpage / false .meta:n = {linktoc=section}
1565     ,linktocpage .default:n = true
1566 }
1567

link (setup key) This booleans allow to disable the link types.
url (setup key) 1568 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1569 {
menu (setup key) 1570     \keys_define:nn { hyp }
run (setup key) 1571     {
1572         ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1573     }
1574 }

1575 \keys_define:nn { hyp }
1576 {
1577     ,baseurl .code:n =

```

```

1578 {
1579   \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1580   \tl_if_empty:NTF \l__hyp_tmpa_tl
1581   {
1582     \pdfmanagement_remove:nn {Catalog} { URI }
1583   }
1584   {
1585     \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1586   }
1587   \__hyp_store_metadata:nn {baseurl}{#1}
1588 }
1589 %only false does something ...
1590 ,bookmarks .choice:
1591 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1592 ,bookmarks / true .code:n = {}
1593 ,bookmarks .default:n = {true}
1594 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1595 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1596 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1597 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1598 ,pdfcenterwindow .choice:
1599 ,pdfcenterwindow / false .code:n =
1600 {
1601   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1602 }
1603 ,pdfcenterwindow / true .code:n =
1604 {
1605   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1606 }
1607 ,pdfcenterwindow / .code:n =
1608 {
1609   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1610 }
1611 ,pdfcenterwindow / unknown .code:n =
1612 {
1613   \msg_warning:nnee { hyp } { no-bool }
1614   { pdfcenterwindow }
1615   { \exp_not:n {#1} }
1616 }
1617 ,pdfcenterwindow .default:n = true
1618 ,pdfdirection .choice:
1619 ,pdfdirection / L2R .code:n =
1620 {
1621   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1622 }
1623 ,pdfdirection / R2L .code:n =
1624 {
1625   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1626 }
1627 ,pdfdirection / .code:n =
1628 {
1629   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1630 }
1631 ,pdfdirection / unknown .code:n =

```

```

1632     {
1633         \msg_warning:nneee { hyp } { unknown-choice+empty }
1634         { pdfdirection }
1635         { L2R , R2L }
1636         { \exp_not:n {#1} }
1637     }
1638 ,pdffitwindow .choice:
1639 ,pdffitwindow / false .code:n =
1640     {
1641         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1642     }
1643 ,pdffitwindow / true .code:n =
1644     {
1645         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1646     }
1647 ,pdffitwindow .default:n = true
1648 ,pdfduplex .choices:nn =
1649     {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1650     {
1651         \pdf_version_compare:NnTF > {1.6}
1652         {
1653             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1654             { PrintDuplex } { /#1 }
1655         }
1656         {
1657             \msg_warning:nnee
1658             {hyp}
1659             {ignore-deprecated-or-unknown-option-in-pdf-version}
1660             {pdfduplex}
1661             {\pdf_version:}
1662         }
1663     }%
1664 ,pdfduplex / .code:n =
1665     {
1666         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1667     }
1668 ,pdfduplex / unknown .code:n =
1669     {
1670         \msg_warning:nneee { hyp } { unknown-choice+empty }
1671         { pdfduplex }
1672         { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1673         { \exp_not:n {#1} }
1674     }
1675 ,pdffitwindow .choice:
1676 ,pdffitwindow / false .code:n =
1677     {
1678         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1679     }
1680 ,pdffitwindow / true .code:n =
1681     {
1682         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1683     }
1684 ,pdffitwindow / .code:n =
1685     {

```

```

1686     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1687   }
1688   ,pdffitwindow .default:n = true
1689   ,pdffitwindow / unknown .code:n =
1690   {
1691     \msg_warning:nnee { hyp } { no-bool }
1692     { pdffitwindow }
1693     { \exp_not:n {#1} }
1694   }
1695   ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1696   ,pdflinkmargin .initial:n = {1pt}
1697   ,pdfmenubar .choice:
1698   ,pdfmenubar / true .code:n =
1699   {
1700     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1701   }
1702   ,pdfmenubar / false .code:n =
1703   {
1704     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1705     { HideMenubar } { true }
1706   }
1707   ,pdfmenubar / .code:n =
1708   {
1709     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1710   }
1711   ,pdfmenubar .default:n = true
1712   ,pdfmenubar / unknown .code:n =
1713   {
1714     \msg_warning:nnee { hyp } { no-bool }
1715     { pdfmenubar }
1716     { \exp_not:n {#1} }
1717   }
1718   ,pdfnewwindow .choice:
1719   ,pdfnewwindow / true .code:n =
1720   {
1721     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1722     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1723   }
1724   ,pdfnewwindow / false .code:n =
1725   {
1726     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1727     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1728   }
1729   ,pdfnewwindow / .code:n =
1730   {
1731     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1732     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1733   }
1734   ,pdfnonfullscreenpagemode .choices:nn =
1735   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1736   {
1737     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1738     { NonFullScreenPageMode } {/#1}
1739   }

```



```

1740 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1741 {
1742     \pdf_version_compare:NnTF < {1.6}
1743     {
1744         %message
1745     }
1746     {
1747         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1748         {NonFullScreenPageMode}{/UseAttachments}
1749     }
1750 }
1751 ,pdfnonfullscreenpagemode / .code:n =
1752 {
1753     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1754 }
1755 ,pdfnonfullscreenpagemode / unknown .code:n =
1756 {
1757     \msg_warning:nnee { hyp } { unknown-choice+empty }
1758     { pdfnonfullscreenpagemode }
1759     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1760     { \exp_not:n {#1} }
1761 }
1762 ,pdfnumcopies .code:n =
1763 {
1764     \pdf_version_compare:NnTF > {1.6}
1765     {
1766         \tl_if_empty:nTF {#1}
1767         {
1768             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1769         }
1770         {
1771             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1772             {NumCopies}{#1}
1773         }
1774     }
1775     {
1776         \msg_warning:nnee
1777         {hyp}
1778         {ignore-deprecated-or-unknown-option-in-pdf-version}
1779         {pdfnumcopies}
1780         {\pdf_version:}
1781     }
1782 }
1783 ,pdfpagelayout .choices:nn =
1784 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1785 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1786 ,pdfpagelayout / .code:n =
1787 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1788 ,pdfpagelayout / unknown .code:n =
1789 {
1790     \msg_warning:nnee { hyp } { unknown-choice+empty }
1791     { pdfpagelayout }
1792     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1793     { \exp_not:n {#1} }

```

```

1794     }
1795 ,pdfpagemode .choices:nn =
1796   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1797   { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1798 ,pdfpagemode / UseAttachments .code:n =
1799   {
1800     \pdf_version_compare:NnTF > {1.5}
1801     {
1802       \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1803     }
1804     {
1805       \msg_warning:nnee
1806         {hyp}
1807         {ignore-deprecated-or-unknown-value-in-pdf-version}
1808         {UseAttachments}
1809         {\pdf_version:}
1810     }
1811   }
1812 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1813 ,pdfpagemode / unknown .code:n =
1814   {
1815     \msg_warning:nnee { hyp } { unknown-choice+empty }
1816     { pdfpagemode }
1817     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1818     { \exp_not:n {#1} }
1819   }
1820 ,pdfpagescrop .code:n =
1821   {
1822     \tl_if_empty:nTF {#1} %or blank?
1823     {
1824       \pdfmanagement_remove:nn {Pages} { CropBox }
1825     }
1826     {
1827       \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1828     }
1829   }
1830 ,pdfpicktraybypdfsize .choice:
1831 ,pdfpicktraybypdfsize / true .code:n =
1832   {
1833     \pdf_version_compare:NnTF > {1.6}
1834     {
1835       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1836         { PickTrayByPDFSize } { true }
1837     }
1838     {
1839       \msg_warning:nnee
1840         {hyp}
1841         {ignore-deprecated-or-unknown-option-in-pdf-version}
1842         {pdfpicktraybypdfsize}
1843         {\pdf_version:}
1844     }
1845   }
1846 ,pdfpicktraybypdfsize / false .code:n =
1847   {

```

```

1848     \pdf_version_compare:NnTF > {1.6}
1849     {
1850         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1851         { PickTrayByPDFSize } { false }
1852     }
1853     {
1854         \msg_warning:nnee
1855         {hyp}
1856         {ignore-deprecated-or-unknown-option-in-pdf-version}
1857         {pdfpicktraybypdfsize}
1858         {\pdf_version:}
1859     }
1860 }
1861 ,pdfpicktraybypdfsize / .code:n =
1862 {
1863     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1864 }
1865 ,pdfpicktraybypdfsize / unknown .code:n =
1866 {
1867     \msg_warning:nnee { hyp } { no-bool }
1868     { picktraybypdfsize }
1869     { \exp_not:n {#1} }
1870 }
1871 ,pdfprintarea .choices:nn =
1872 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1873 {
1874     \pdf_version_compare:NnTF < {2.0}
1875     {
1876         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1877         { PrintArea } { /#1 }
1878     }
1879     {
1880         \msg_warning:nnee
1881         {hyp}
1882         {ignore-deprecated-or-unknown-option-in-pdf-version}
1883         {pdfprintarea}
1884         {\pdf_version:}
1885     }
1886 }%
1887 ,pdfprintarea / .code:n =
1888 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1889 ,pdfprintarea / unknown .code:n =
1890 {
1891     \msg_warning:nneee { hyp } { unknown-choice+empty }
1892     { pdfprintarea }
1893     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1894     { \exp_not:n {#1} }
1895 }
1896 ,pdfprintclip .choices:nn =
1897 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1898 {
1899     \pdf_version_compare:NnTF < {2.0}
1900     {
1901         \pdfmanagement_add:nne {Catalog / ViewerPreferences }

```

```

1902         { PrintClip } { /#1 }
1903     }
1904     {
1905         \msg_warning:nnee
1906         {hyp}
1907         {ignore-deprecated-or-unknown-option-in-pdf-version}
1908         {pdfprintclip}
1909         {\pdf_version:}
1910     }
1911 }%
1912 ,pdfprintclip / .code:n =
1913 {
1914     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1915 }
1916 ,pdfprintclip / unknown .code:n =
1917 {
1918     \msg_warning:nneee
1919     { hyp }
1920     { unknown-choice+empty }
1921     { pdfprintclip }
1922     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1923     { \exp_not:n {#1} }
1924 }
1925 ,pdfprintpagerange .code:n =
1926 {
1927     \pdf_version_compare:NnTF > {1.6}
1928     {
1929         \tl_if_empty:nTF { #1}
1930         {
1931             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1932             { PrintPageRange }
1933         }
1934         {
1935             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1936             {PrintPageRange}{[#1]}
1937         }
1938     }
1939     {
1940         \msg_warning:nnee
1941         {hyp}
1942         {ignore-deprecated-or-unknown-option-in-pdf-version}
1943         {pdfprintpagerange}
1944         {\pdf_version:}
1945     }
1946 }
1947 ,pdfprintscaling .choices:nn =
1948 { None, AppDefault }
1949 {
1950     \pdf_version_compare:NnTF > {1.5}
1951     {
1952         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1953         { PrintScaling } { /#1 }
1954     }
1955     {

```

```

1956         \msg_warning:nnee
1957         {hyp}
1958         {ignore-deprecated-or-unknown-option-in-pdf-version}
1959         {pdfprintscaling}
1960         {\pdf_version:}
1961     }
1962 }%
1963 ,pdfprintscaling / .code:n =
1964 {
1965     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1966 }
1967 ,pdfprintscaling / unknown .code:n =
1968 {
1969     \msg_warning:nneee { hyp } { unknown-choice+empty }
1970     { pdfprintarea }
1971     { None, AppDefault }
1972     { \exp_not:n {#1} }
1973 }
1974 ,pdfremotestartview .code:n =
1975 {
1976     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1977     \exp_args:NNV
1978     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1979     {
1980         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1981     }
1982     {
1983         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1984         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1985     }
1986 }
1987 ,pdfremotestartview .initial:n = {Fit}
1988 % pdfstartpage is special as it shares code with pdfstartview
1989 ,pdfstartpage .code:n =
1990 {
1991     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1992     \bool_if:nTF
1993     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1994     {
1995         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1996     }
1997     {
1998         \pdfmanagement_add:nne {Catalog} { OpenAction }
1999         {
2000             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2001         }
2002     }
2003 }
2004 ,pdfstartpage .initial:n =1
2005 ,pdfstartview .code:n =
2006 {
2007     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
2008     \exp_args:NNV
2009     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_

```

```

2010     {
2011     \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
2012     }
2013     {
2014     \msg_warning:nmmn {hyp}{invalid-destination-value}{#1}{pdfstartview}
2015     \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2016     }
2017     \bool_if:nTF
2018     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2019     {
2020     \pdfmanagement_remove:nn {Catalog} { OpenAction }
2021     }
2022     {
2023     \pdfmanagement_add:nne {Catalog} { OpenAction }
2024     {
2025     [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2026     }
2027     }
2028     }
2029     ,pdfstartview .initial:n = Fit
2030     ,pdftoolbar .choice:
2031     ,pdftoolbar / true .code:n =
2032     {
2033     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2034     }
2035     ,pdftoolbar / false .code:n =
2036     {
2037     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2038     { HideToolbar } { true }
2039     }
2040     ,pdftoolbar / true .code:n =
2041     {
2042     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2043     }
2044     ,pdftoolbar .default:n = true
2045     ,pdftoolbar / unknown .code:n =
2046     {
2047     \msg_warning:nnee { hyp } { no-bool }
2048     { pdftoolbar }
2049     { \exp_not:n {#1} }
2050     }
2051     % pdfview see below.
2052     ,pdfviewarea .choices:nn =
2053     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2054     {
2055     \pdf_version_compare:NnTF < {2.0}
2056     {
2057     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2058     { ViewArea } { /#1 }
2059     }
2060     {
2061     \msg_warning:nnee
2062     {hyp}
2063     {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

2064         {pdfviewarea}
2065         {\pdf_version:}
2066     }
2067 }%
2068 ,pdfviewarea / .code:n =
2069 {
2070     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2071 }
2072 ,pdfviewarea / unknown .code:n =
2073 {
2074     \msg_warning:nnee { hyp } { unknown-choice+empty }
2075     { pdfviewarea }
2076     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2077     { \exp_not:n {#1} }
2078 }
2079 ,pdfviewclip .choices:nn =
2080 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2081 {
2082     \pdf_version_compare:NnTF < {2.0}
2083     {
2084         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2085         { ViewClip } { /#1 }
2086     }
2087     {
2088         \msg_warning:nnee
2089         {hyp}
2090         {ignore-deprecated-or-unknown-option-in-pdf-version}
2091         {pdfviewclip}
2092         {\pdf_version:}
2093     }
2094 }%
2095 ,pdfviewclip / .code:n =
2096 {
2097     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2098 }
2099 ,pdfviewclip / unknown .code:n =
2100 {
2101     \msg_warning:nnee { hyp } { unknown-choice+empty }
2102     { pdfviewclip }
2103     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2104     { \exp_not:n {#1} }
2105 }
2106 ,pdfwindowui .choice:
2107 ,pdfwindowui / true .code:n =
2108 {
2109     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2110 }
2111 ,pdfwindowui / false .code:n =
2112 {
2113     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2114     { HideWindowUI } { true }
2115 }
2116 ,pdfwindowui / .code:n =
2117 {

```

```

2118     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2119   }
2120   ,pdfwindowui / unknown .code:n =
2121   {
2122     \msg_warning:nnee { hyp } { no-bool }
2123     { pdfwindowui }
2124     { \exp_not:n {#1} }
2125   }
2126   ,pdfwindowui .default:n = true
2127 }

```

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

```

2128 \keys_define:nn { hyp }
2129 {
2130   ,pdfview .code:n =
2131   {
2132     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2133     \str_case_e:nmF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2134     {
2135       { xyz }
2136       {
2137         \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2138         {
2139           \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2140           \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2141           {
2142             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2143           }
2144           {
2145             \tl_set:Ne \l__hyp_dest_pdfview_tl
2146             {
2147               \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2148             }
2149           }
2150         }
2151         {
2152           \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2153         }
2154       }
2155       { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2156       { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2157       { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2158       { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2159       { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2160       { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2161       { fitr }
2162       {
2163         \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2164         {
2165           \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2166         }
2167         {
2168           %ensure 4 values ...
2169           \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}

```



```

2170         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2171         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2172         \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2173         \hbox_set_to_wd:Nnn \l__hyp_dest_box
2174         {
2175             \fp_eval:n
2176             {
2177                 round
2178                 (
2179                     abs
2180                     (
2181                         \seq_item:Nn\l__hyp_tmpa_seq{4}
2182                         -
2183                         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2184                     ),
2185                     3
2186                 )
2187             }bp
2188         }{}
2189         \box_set_dp:Nn \l__hyp_dest_box
2190         {
2191             \fp_eval:n
2192             {
2193                 round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2194             }bp
2195         }
2196         \box_set_ht:Nn \l__hyp_dest_box
2197         {
2198             \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2199         }
2200     }
2201 }
2202 {
2203 {
2204     \msg_warning:nmmn {hyp}{invalid-destination-value}{#1}{pdfview}
2205     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2206 }
2207 }
2208 ,pdfview .initial:n = {xyz}
2209 }

```

### 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.

```

2210 \keys_define:nn { hyp }
2211 {
2212     ,pdflang .code:n =
2213     {
2214         \tl_if_empty:nF { #1 }
2215         {
2216             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }

```

```

2217         \AddToDocumentProperties[document]{lang}{#1}
2218     }
2219 }
2220 }

```

### 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*) clist and use only the first value. The values are expanded with `\text_expand:n`  
**pdfsubject** (*setup key*)  
**pdfproducer** (*setup key*)  
**pdfkeywords** (*setup key*)

```

2221 \regex_new:N\l__hyp_optlang_regex
2222 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\]]\ (.*)}
2223 \cs_generate_variant:Nn\clist_item:nn{on}
2224 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2225 {
2226   \keys_define:nn { hyp }
2227   {
2228     pdf#1 .code:n =
2229     {
2230       \tl_set:Ne\l__hyp_tmpa_tl {\text_expand:n{##1}}
2231       \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2232       \tl_if_empty:NTF \l__hyp_tmpa_tl
2233       {
2234         \str_case:nnF { #1 }
2235         {
2236           {creator}
2237           {
2238             \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2239             \pdfmanagement_add:nne {Info}{Creator}{()}
2240           }
2241           {producer}
2242           {
2243             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2244             \pdfmanagement_add:nne {Info}{Producer}{()}
2245           }
2246         }
2247         {
2248           \pdfmanagement_remove:nn {Info}{#2}
2249         }
2250       }
2251     }
2252     \tl_set:Ne\l__hyp_tmpb_tl {\clist_item:on{\l__hyp_tmpa_tl}{1}}
2253     \exp_args:NNV
2254     \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_s
2255     \seq_if_empty:NTF\l__hyp_tmpa_seq
2256     {
2257       \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2258     }
2259     {
2260       \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2261     }
2262     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}

```

```

2263         {
2264             \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2265         }
2266     }
2267 }
2268 }
2269 \keys_define:nn { hyp / info }
2270 {
2271     #2 .code:n =
2272     {
2273         \tl_set:Nel__hyp_tmpa_tl {\text_expand:n{##1}}
2274         \__hyp_store_metadata:eo {pdf\str_lowercase:n{#1}}{\l__hyp_tmpa_tl}
2275         \tl_if_blank:nTF {##1}
2276         {
2277             \pdfmanagement_remove:nn {Info}{#2}
2278         }
2279         {
2280             \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2281             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2282             {
2283                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2284             }
2285         }
2286     }
2287     ,unknown .code:n =
2288     {
2289         \__hyp_text_pdfstring_info:eN {##1}\l__hyp_tmpa_str
2290         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2291         {
2292             \exp_args:Nno
2293             \pdfmanagement_add:nne {Info}
2294             { \l_keys_key_str } {\l__hyp_tmpa_str}
2295         }
2296     }
2297 }
2298 }
2299 \__hyp_setup_info_key:nn {author} {Author}
2300 \__hyp_setup_info_key:nn {title} {Title}
2301 \__hyp_setup_info_key:nn {producer} {Producer}
2302 \__hyp_setup_info_key:nn {creator} {Creator}
2303 % ignored key: addtopdfcreator
2304 \__hyp_setup_info_key:nn {subject} {Subject}
2305 \__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)
2306 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2307 {
2308     \keys_define:nn { hyp }
2309     {
2310         pdf#1 .code:n =
2311         {
2312             \tl_if_blank:nTF {##1}
2313             {

```

```

2314         \pdfmanagement_remove:nn {Info}{#2}
2315     }
2316     {
2317         \pdfmanagement_add:nne {Info}{#2}{##1}
2318     }
2319     \__hyp_store_metadata:nn {pdf#1}{##1}
2320     \AddToDocumentProperties[document]{#1}{##1}
2321 }
2322 }
2323 \keys_define:nn { hyp / info }
2324 {
2325     #2 .code:n =
2326     {
2327         \tl_if_blank:nTF {##1}
2328         {
2329             \pdfmanagement_remove:nn {Info}{#2}
2330         }
2331         {
2332             \pdfmanagement_add:nne {Info}{#2}{##1}
2333         }
2334         \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2335     }
2336 }
2337 }
2338
2339 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2340 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2341 \keys_define:nn { hyp }
2342 {
2343     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2344 }

```

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

```

2345 \keys_define:nn { hyp }
2346 {
2347     ,pdftrapped .code:n =
2348     {
2349         \exp_args:Nne
2350         \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2351     }
2352     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2353     {
2354         \pdfmanagement_add:nne {Info}{Trapped}
2355         {/
2356             \str_uppercase:f { \str_head:n { #1 } }
2357             \str_lowercase:f { \str_tail:n { #1 } }
2358         }
2359         \__hyp_store_metadata:ne {pdftrapped}
2360         {
2361             \str_uppercase:f { \str_head:n { #1 } }
2362             \str_lowercase:f { \str_tail:n { #1 } }
2363         }
2364     }
2365     ,_pdftrapped / unknown .code:n =

```

```

2366     {
2367     \msg_warning:nneee { hyp } { unknown-choice }
2368     { pdftrapped }
2369     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2370     { \exp_not:n {#1} }
2371     }
2372 }

```

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

```

2373 \keys_define:nn { hyp }
2374 {
2375   pdfinfo .code:n =
2376   {
2377     \keys_set:nn { hyp / info } { #1 }
2378   }
2379 }

```

Now we set some default values

```

2380 \keys_set:nn { hyp} {pdfcreator = LaTeX-with-hyperref}
2381 \keys_set:nn { hyp} {pdfauthor = }
2382 \keys_set:nn { hyp} {pdftitle = }
2383 \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:

```

2384 \clist_map_inline:nn
2385 {
2386   ,pdfcopyright
2387   ,pdftype
2388   ,pdflicenseurl
2389   ,pdfauthortitle
2390   ,pdfcaptionwriter
2391   ,pdfmetalang
2392   ,pdfsource
2393   ,pdfdocumentid
2394   ,pdfinstanceid
2395   ,pdfversionid
2396   ,pdfrendition
2397   ,pdfpublication
2398   ,pdfpubtype
2399   ,pdfbytes
2400   ,pdfnumpages
2401   ,pdfissn
2402   ,pdfeissn
2403   ,pdfisbn
2404   ,pdfbookedition
2405   ,pdfpublisher
2406   ,pdfvolumenum
2407   ,pdfissuenum
2408   ,pdfpagerange
2409   ,pdfdoi

```

```

2410 ,pdfurl
2411 ,pdfidentifier
2412 ,pdfsubtitle
2413 ,pdfpubstatus
2414 ,pdfcontactaddress
2415 ,pdfcontactcity
2416 ,pdfcontactregion
2417 ,pdfcontactpostcode
2418 ,pdfcontactcountry
2419 ,pdfcontactphone
2420 ,pdfcontactemail
2421 ,pdfcontacturl
2422 ,pdfdate
2423 }
2424 {
2425 \keys_define:nn { hyp }
2426 {
2427 #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2428 }
2429 }
2430

```

## 12.5 Transitions

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

```

2431 \keys_define:nn { hyp }
2432 {
2433 pdfpageduration .code:n =
2434 {
2435 \tl_if_blank:nTF { #1 }
2436 {
2437 \pdfmanagement_remove:nn {Page}{Dur}
2438 }
2439 {
2440 \pdfmanagement_add:nnn {Page}{Dur}{#1}
2441 }
2442 }
2443 }

```

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

```

2444 \keys_define:nn { hyp }
2445 {
2446   pdfpagetransition .code:n =
2447   {
2448     \tl_if_blank:nTF {#1}
2449     {
2450       \pdfmanagement_remove:nn {Page}{Trans}
2451     }
2452     {
2453       \group_begin:
2454       \keys_set:nn { hyp / trans }{style=R,#1}
2455       \pdf_object_unnamed_write:ne { dict }
2456       {
2457         \pdfdict_use:n {l__hyp_page/Trans}
2458       }
2459       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2460       \group_end:
2461     }
2462   }
2463 }
2464 \keys_define:nn { hyp / trans }
2465 {
2466   ,style .choices:nn =
2467   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2468   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2469   ,style / unknown .code:n =
2470   {
2471     \msg_warning:nnee { hyp } { unknown-choice }
2472     { trans / style }
2473     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2474     { \exp_not:n {#1} }
2475   }
2476   ,duration .code:n =
2477   {
2478     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
  
```

```

2479     }
2480 ,direction .choices:nn =
2481     {H,V}
2482     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2483 ,direction .choices:nn =
2484     {0,90,180,270,315}
2485     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2486 ,direction / None .code:n =
2487     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2488 ,direction / unknown .code:n =
2489     {
2490         \msg_warning:nneee { hyp } { unknown-choice }
2491         { trans / direction }
2492         {
2493             H~(horizontal,~only~Split,~Blinds),
2494             V~(vertical,~only~Split,~Blinds),
2495             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2496             90~(bottom~to~top,~only~Wipe),
2497             180~(right~to~left,~only~Wipe),
2498             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2499             315~(top~left~to~bottom,~only~Glitter),
2500             None~(only~Fly)
2501         }
2502         { \exp_not:n {#1} }
2503     }
2504 ,motion .choices:nn =
2505     {I,0}
2506     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2507 ,motion / unknown .code:n =
2508     {
2509         \msg_warning:nneee { hyp } { unknown-choice }
2510         { trans / motion }
2511         { I~(inwards) , 0~(outwards) }
2512         { \exp_not:n {#1} }
2513     }
2514 ,scale .code:n =
2515     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2516 ,opaque .choices:nn = {true,false}
2517     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2518 ,opaque / unknown .code:n =
2519     {
2520         \msg_warning:nneee { hyp } { unknown-choice }
2521         { trans / B }
2522         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2523         { \exp_not:n {#1} }
2524     }
2525 % try to set unknown keys as style
2526 ,unknown .code:n =
2527     {
2528         % warning ...
2529         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2530     }
2531 }

```



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

```
2532 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
    Unfinished Form field code
2533 \NewDocumentCommand \MakeFieldObject { m m }
2534 {
2535   \pdfxform_new:nnn { #2 }{-} { #1 }
2536 }
2537
2538
2539 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2540 \prop_new:N \g__hyp_AcroForm_Fields_prop
2541
2542 \let\HyField@afields\@empty
2543 \let\HyField@cofields\@empty
2544 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2545
2546 % the value doesn't matter, but with a prop we avoid duplicates and it is
2547 % clearly faster than removing them from a sequence
2548 \def\HyField@AuxAddToFields#1
2549 {
2550   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2551 }%
2552
2553 %fields with empty key get a value too -- lets hope that
2554 %this give the expected behaviour
2555 \def\HyField@AuxAddToCoFields #1 #2
2556 {
2557   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2558 }
2559
2560 \Hy@AtBeginDocument
2561 {
2562   \if@filesw
2563     \immediate\write\@mainaux{%
2564       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2565     }%
2566     \immediate\write\@mainaux{%
2567       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2568     }%
2569   \fi
2570   \let\HyField@AfterAuxOpen\@firstofone
2571 }%
2572
2573 \def\HyField@AddToFields
2574 {
2575   \exp_args:N\HyField__hypAddToFields
2576   {
2577     \pdfannot_box_ref_last:
2578   }
2579   \ifx\Fld@calculate@code\@empty
2580   \else
2581     \begingroup
2582     \Hy@safe@activestru
```

```

2583     \edef\Hy@temp{%
2584         \endgroup
2585         \if@filesw
2586             \write\@mainaux
2587             {
2588                 \string\HyField@AuxAddToCoFields
2589                 {
2590                     \Fld@calculate@sortkey
2591                 }
2592                 {
2593                     \pdfannot_box_ref_last:
2594                 }
2595             }
2596         \fi
2597     }%
2598 \Hy@temp
2599 \fi
2600 }%
2601
2602 \def\HyField__hypAddToFields#1{
2603 \HyField@AfterAuxOpen{%
2604 \if@filesw
2605 \write\@mainaux{%
2606 \string\HyField@AuxAddToFields{#1}%
2607 }%
2608 \fi
2609 }%
2610 }%
2611
2612 \ExplSyntaxOff
2613 \ExplSyntaxOn
2614
2615 \def\@Form[#1]
2616 {
2617     \kvsetkeys{Form}{#1}
2618     \pdf@ifdraftmode{}
2619     {
2620         \Hy@FormObjects
2621         \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2622         {
2623             \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2624             %\pdfmanagement_show:n { Catalog / AcroForm }
2625         }
2626         \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2627         {
2628             \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2629             {
2630                 \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2631             }
2632             \seq_sort:Nn \l__hyp_tmpa_seq
2633             {
2634                 \str_compare:nNnTF {##1} > {##2}
2635                 { \sort_return_swapped: }
2636                 { \sort_return_same: }

```

```

2637     }
2638     \seq_map_inline:Nn \l__hyp_tmpa_seq
2639     {
2640         \pdfmanagement_add:nne { Catalog / AcroForm }
2641         { CO }
2642         {
2643             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2644         }
2645     }
2646 }
2647 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2648 {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2649 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2650 {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2651 \pdfmanagement_add:nne {Catalog /AcroForm}
2652 {DA}{(/Helv~10~Tf~0~g)}
2653 \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2654 {
2655     \legacy_if:nT { HyField@NeedAppearances }
2656     {
2657         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2658     }
2659 }
2660 {
2661     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2662 }
2663 }
2664 }
2665 \ExplSyntaxOff
2666 \let\@endForm\@empty
2667 \let\HyAnn@AbsPageLabel\@empty
2668 \let\Fld@pageobjref\@empty
2669
2670 \ExplSyntaxOn
2671 \newcount\HyAnn@Count
2672 \HyAnn@Count=\z@
2673 \def\HyAnn@AbsPageLabel
2674 {
2675     \global\advance\HyAnn@Count by\@ne
2676     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2677     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2678     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2679     \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2680     \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2681 }%
2682 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2683 \def\Fld@pageobjref
2684 {
2685     \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2686     {
2687         /P~\pdf_pageobject_ref:e
2688         {
2689             \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2690         }
2691     }

```

```

2691     }
2692   }
2693   \ExplSyntaxOff
2694   \ExplSyntaxOn
2695   %% check if the attr should be set through
2696   %% hooks.
2697   %% check if options are missing.
2698   \def\@TextField[#1]#2{% parameters, label
2699     \def\Fld@name{#2}%
2700     \let\Fld@default\@empty
2701     \let\Fld@value\@empty
2702     \def\Fld@width{\DefaultWidthofText}%
2703     \def\Fld@height{%
2704       \ifFld@multiline
2705         \DefaultHeightofTextMultiline
2706       \else
2707         \DefaultHeightofText
2708       \fi
2709     }%
2710     \begingroup
2711     \expandafter\HyField@SetKeys\expandafter{%
2712       \DefaultOptionsofText,#1%
2713     }%
2714     \PDFForm@Name
2715     \HyField@FlagsText
2716     \ifFld@hidden\def\Fld@width{1sp}\fi
2717     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2718     \LayoutTextField{#2}{%
2719       \leavevmode
2720       \HyAnn@AbsPageLabel
2721       \Hy@escapeform\PDFForm@Text
2722       \pdfannot_box:nmmn
2723         {\Fld@width}
2724         {\Fld@height}
2725         {Opt} %is this correct?
2726         {\PDFForm@Text}
2727       \MakeTextField{\Fld@width}{\Fld@height}
2728       \HyField@AddToFields
2729     }%
2730   \endgroup
2731 }
2732 \providecommand\@curropt{}
2733 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2734   \def\Fld@name{#2}
2735   \let\Fld@default\relax
2736   \let\Fld@value\relax
2737   \def\Fld@width{\DefaultWidthofChoiceMenu}
2738   \def\Fld@height{\DefaultHeightofChoiceMenu}
2739   \begingroup
2740     \Fld@menulength=0 %
2741     \@tempdima\z@
2742     \clist_map_variable:nNn { #3 } \@curropt
2743     %\@for\@curropt:=#3\do
2744     {%

```

```

2745     \expandafter\Fld@checkequals\@curropt==\%
2746     \Hy@StepCount\Fld@menulength
2747     \settowidth{\@tempdimb}{\@currDisplay}%
2748     \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2749   }%
2750 \advance\@tempdima by~15\p@
2751 \beginingroup
2752   \HyField@SetKeys{#1}
2753 \edef\x{\endgroup
2754   \noexpand\expandafter
2755   \noexpand\HyField@SetKeys
2756   \noexpand\expandafter{%
2757     \expandafter\noexpand\csname DefaultOptionsof%
2758     \ifFld@radio
2759       Radio%
2760     \else
2761       \ifFld@combo
2762         \ifFld@popdown
2763           PopdownBox%
2764         \else
2765           ComboBox%
2766         \fi
2767       \else
2768         ListBox%
2769       \fi
2770     \fi
2771   \endcsname
2772   }%
2773 }\x
2774 \HyField@SetKeys{#1}%
2775 \PDFForm@Name
2776 \ifFld@hidden\def\Fld@width{1sp}\fi
2777 \ifx\Fld@value\relax
2778   \let\Fld@value\Fld@default
2779 \fi
2780 \LayoutChoiceField{#2}{%
2781   \ifFld@radio
2782     \HyField@FlagsRadioButton
2783     \__hypRadio{#3}%
2784   \else
2785     \beginingroup
2786     \HyField@FlagsChoice
2787     \ifdim\Fld@width<\@tempdima
2788       \ifdim\@tempdima<1cm\@tempdima1cm\fi
2789       \edef\Fld@width{\the\@tempdima}%
2790     \fi
2791     \ifFld@combo
2792     \else
2793       \@tempdima=\the\Fld@menulength\Fld@charsize
2794       \advance\@tempdima by~\Fld@borderwidth bp %
2795       \advance\@tempdima by~\Fld@borderwidth bp %
2796       \edef\Fld@height{\the\@tempdima}%
2797     \fi
2798     \__hypListbox{#3}%

```

```

2799         \endgroup
2800     \fi
2801 }%
2802 \endgroup
2803 }
2804
2805 \def\__hypRadio#1{%
2806     \Fld@listcount=0-%
2807     %\show\Fld@default
2808     \EdefEscapeName\Fld@default{\Fld@default}%
2809     \clist_map_variable:nNn { #1 } \@curropt
2810     %\@for\@curropt:=#1\do
2811     {%
2812         \expandafter\Fld@checkequals\@curropt==\%
2813         \EdefEscapeName\@currValue{\@currValue}%
2814         \Hy@StepCount\Fld@listcount
2815         \@currDisplay\space
2816         \leavevmode
2817         \HyAnn@AbsPageLabel
2818         \Hy@escapeform\PDFForm@Radio
2819         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2820         {
2821             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2822             {
2823                 \group_begin:
2824                 \fontfamily{pzd}
2825                 \fontencoding{U}
2826                 \fontseries{m}
2827                 \fontshape{n}
2828                 \selectfont
2829                 \char123
2830                 \group_end:
2831             }
2832         }
2833         \pdfannot_box:nnne
2834         {\Fld@width}
2835         {\Fld@height}
2836         {Opt} %is this correct?
2837         {
2838             \PDFForm@Radio
2839             /AP
2840             <<
2841             /N
2842             <<
2843             /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2844             %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2845             >>
2846             >>
2847         }
2848         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2849         \int_compare:nNnT { \Fld@listcount } = { 1 }
2850         { \HyField@AddToFields }
2851         \c_space_tl % deliberate space between radio buttons
2852         % to do: --> should be configurable

```

```

2853 }%
2854 }
2855
2856 \newcount\Fld@listcount
2857 \def\_hypListBox#1
2858 {
2859   \HyField@PDFChoices{#1}
2860   \mode_leave_vertical:
2861   \HyAnn@AbsPageLabel
2862   \Hy@escapeform\PDFForm@List
2863   \pdf_link_user:nnn
2864   {widget} %perhaps we need more types??
2865   {\PDFForm@List}
2866   {\MakeChoiceField{\Fld@width}{\Fld@height}}
2867   \HyField@AddToFields
2868 }
2869
2870
2871 \def\@PushButton[#1]#2{% parameters, label
2872   \def\Fld@name{#2}%
2873   \group_begin:
2874     \exp_args:No\HyField@SetKeys
2875     {
2876       \DefaultOptionsofPushButton,#1
2877     }
2878     \PDFForm@Name
2879     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2880     {
2881       \HyField@FlagsPushButton
2882       \legacy_if:nT {Fld@hidden}
2883       {
2884         \def\Fld@width{1sp}
2885       }
2886       \LayoutPushButtonField
2887       {
2888         \mode_leave_vertical:
2889         \HyAnn@AbsPageLabel
2890         \Hy@escapeform\PDFForm@Push
2891         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2892         \pdfannot_box:nnnn
2893         {\box_wd:N\l_tmpa_box}
2894         {\box_ht:N\l_tmpa_box}
2895         {\box_dp:N\l_tmpa_box} %is this correct?
2896         {\PDFForm@Push}
2897         {\box_use:N\l_tmpa_box}
2898         \HyField@AddToFields
2899       }
2900     }
2901     {
2902       \msg_error:nn { hyp }{ pdfa-no-push-button }
2903       \LayoutPushButtonField
2904       {
2905         \mode_leave_vertical:
2906         \MakeButtonField{#2}

```

```

2907     }
2908   }
2909   \group_end:
2910 }
2911
2912 \def\@Submit[#1]#2
2913 {
2914   \def\Fld@width {\DefaultWidthofSubmit}
2915   \def\Fld@height{\DefaultHeightofSubmit}
2916   \group_begin:
2917     \exp_args:No\HyField@SetKeys
2918     {
2919       \DefaultOptionsofSubmit,#1
2920     }
2921   \HyField@FlagsPushButton
2922   \HyField@FlagsSubmit
2923   \legacy_if:nT { Fld@hidden }
2924   {
2925     \def\Fld@width{1sp}
2926   }
2927   \mode_leave_vertical:
2928   \HyAnn@AbsPageLabel
2929   \Hy@escapeform\PDFForm@Submit
2930   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2931   \pdfxform_if_exist:nF
2932   { __hyp_xform_Submit }
2933   {
2934     \pdfxform_new:nnn { __hyp_xform_Submit }{}
2935     {
2936       \fbox{\color_select:n{yellow}\textsf{Submit}}
2937     }
2938     \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2939     {
2940       \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2941     }
2942   }
2943   \pdfannot_box:nnnn
2944   {\box_wd:N\l_tmpa_box}
2945   {\box_ht:N\l_tmpa_box}
2946   {\box_dp:N\l_tmpa_box} %is this correct?
2947   {
2948     \PDFForm@Submit
2949     /AP<<
2950     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2951     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2952     >>
2953   }
2954   \HyField@AddToFields
2955   \box_use:N\l_tmpa_box
2956
2957   \group_end:
2958 }
2959
2960 \def\@Reset[#1]#2

```



```

2961 {
2962   \def\Fld@width {\DefaultWidthofReset}
2963   \def\Fld@height{\DefaultHeightofReset}
2964   \group_begin:
2965     \exp_args:No\HyField@SetKeys
2966       {
2967         \DefaultOptionsofReset,#1
2968       }
2969     \mode_leave_vertical:
2970     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2971     {
2972       \HyField@FlagsPushButton
2973       \legacy_if:nT { Fld@hidden }
2974         { \def\Fld@width{1sp} }
2975       \HyAnn@AbsPageLabel
2976       \Hy@escapeform\PDFForm@Reset
2977       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2978       \pdfannot_box:nnnn
2979         {\box_wd:N\l_tmpa_box}
2980         {\box_ht:N\l_tmpa_box}
2981         {\box_dp:N\l_tmpa_box} %is this correct?
2982         { \PDFForm@Reset }
2983       \HyField@AddToFields
2984       \box_use:N \l_tmpa_box
2985     }
2986     {
2987       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2988       \MakeButtonField{#2}
2989     }
2990   \group_end:
2991 }
2992
2993 \def\@CheckBox[#1]#2
2994 {% parameters, label
2995   \def\Fld@name{#2}
2996   \def\Fld@default{0}
2997   \group_begin:
2998     \def\Fld@width {\DefaultWidthofCheckBox}
2999     \def\Fld@height{\DefaultHeightofCheckBox}
3000     \exp_args:No\HyField@SetKeys
3001       {
3002         \DefaultOptionsofCheckBox,#1
3003       }
3004     \PDFForm@Name
3005     \HyField@FlagsCheckBox
3006     \legacy_if:nT { Fld@hidden }
3007       {
3008         \def\Fld@width{1sp}
3009       }
3010     \LayoutCheckField{#2}
3011     {
3012       \mode_leave_vertical:
3013       \HyAnn@AbsPageLabel
3014       \Hy@escapeform\PDFForm@Check

```

```

3015     \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3016     {
3017         \pdfxform_new:nnn
3018         {__hyp_xform_CheckMarkYes}{-}
3019         {
3020             \group_begin:
3021             \fontfamily{pzd}
3022             \fontencoding{U}
3023             \fontseries{m}
3024             \fontshape{n}
3025             \selectfont
3026             \char51
3027             \group_end:
3028         }
3029         \pdfxform_new:nnn
3030         {__hyp_xform_CheckMarkOff}{-}
3031         {
3032             \group_begin:
3033             \fontfamily{pzd}
3034             \fontencoding{U}
3035             \fontseries{m}
3036             \fontshape{n}
3037             \selectfont
3038             \phantom{\char51} %perhaps xetex needs some small glyph ..
3039             \group_end:
3040         }
3041     }
3042     \pdfannot_box:nnnn
3043     {\Fld@width}
3044     {\Fld@height}
3045     {Opt} %is this correct?
3046     {\PDFForm@Check}
3047     \HyField@AddToFields %check if this works with xelatex ...
3048 }
3049 \group_end:
3050 }
3051 \ExplSyntaxOff
3052
3053 %hm. Should a luatex driver use type1 fonts in fields????
3054 \ExplSyntaxOn
3055 \def\Hy@FormObjects
3056 {
3057     \pdf_object_new:n {__hyp/Encoding/pdfdoc }
3058     \pdf_object_new:n {__hyp/Font/ZaDb }
3059     \pdf_object_new:n {__hyp/Font/Helv }
3060     \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3061     {
3062         /Type/Encoding
3063         /Differences[
3064             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3065             /ring/tilde
3066             \c_space_tl
3067             39/quotesingle
3068             \c_space_tl

```

```

3069         96/grave %
3070         \iow_newline:
3071         128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3072         /fraction/guilsinglleft/guilsinglright/minus/perthousand
3073         /quotedblbase/quotedblleft/quotedblright/quoteleft
3074         /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3075         /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3076         \iow_newline:
3077         164/currency
3078         \c_space_tl
3079         166/brokenbar
3080         \c_space_tl
3081         168/dieresis/copyright/ordfeminine
3082         \c_space_tl
3083         172/logicalnot/.notdef/registered/macron/degree/plusminus
3084         /twosuperior/threesuperior/acute/mu
3085         \c_space_tl
3086         183/periodcentered/cedilla/onesuperior/ordmasculine
3087         \c_space_tl
3088         188/onequarter/onehalf/threequarters
3089         \iow_newline:
3090         192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3091         /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3092         /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3093         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3094         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3095         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3096         /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3097         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3098         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3099         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3100     ]
3101 }
3102 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3103 {
3104     /Type/Font
3105     /Subtype/Type1
3106     /Name/ZaDb
3107     /BaseFont/ZapfDingbats
3108 }
3109 \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3110 {
3111     /Type/Font
3112     /Subtype/Type1
3113     /Name/Helv
3114     /BaseFont/Helvetica
3115     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3116 }
3117 \global\let\Hy@FormObjects\relax
3118 }
3119 \ExplSyntaxOff
3120 \providecommand*{\Fld@pageobjref}{-}
3121 \ifcsname pdf@escapestring\endcsname
3122 \def\Hy@escapeform#1{%

```

```

3123     \ifHy@pdfescapeform
3124     \let\Hy@escapestring\pdfescapestring
3125     \else
3126     \let\Hy@escapestring\@firstofone
3127     \fi
3128   }%
3129   \Hy@escapeform{}}%
3130 \else
3131   \let\Hy@escapestring\@firstofone
3132   \def\Hy@escapeform#1{%
3133     \ifHy@pdfescapeform
3134     \def\Hy@escapestring##1{%
3135       \noexpand\Hy@escapestring{\noexpand##1}%
3136     }%
3137     \edef\Hy@temp{#1}%
3138     \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{ }\@nil
3139     \def\Hy@escapestring##1{%
3140       \@ifundefined{Hy@esc@\string##1}{%
3141         ##1%
3142         \ThisShouldNotHappen
3143       }{%
3144         \csname Hy@esc@\string##1\endcsname
3145       }%
3146     }%
3147     \else
3148     \let\Hy@escapestring\@firstofone
3149     \fi
3150   }%
3151   \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3152     \ifx\#3\%
3153     \else
3154     \expandafter
3155     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3156     \Hy@ReturnAfterFi{%
3157     \Hy__hypescapeform#3\@nil
3158     }%
3159     \fi
3160   }%
3161 \fi
3162 \def\PDFForm@Name{%
3163   \PDFForm__hypName\Fld@name
3164   \ifx\Fld@altname\relax
3165   \else
3166   \PDFForm__hypName\Fld@altname
3167   \fi
3168   \ifx\Fld@mappingname\relax
3169   \else
3170   \PDFForm__hypName\Fld@mappingname
3171   \fi
3172 }
3173 \def\PDFForm__hypName#1{%
3174   \begingroup
3175   \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3176   \ifHy@unicode

```

```

3177         \Hy@unicodedefalse
3178         \fi
3179         \fi
3180         \pdfstringdef\Hy@gtemp#1%
3181     \endgroup
3182     \let#1\Hy@gtemp
3183 }
3184 \def\Fld@X@additionalactions{%
3185     \ifx\Fld@keystroke@code\@empty
3186     \else
3187         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3188     \fi
3189     \ifx\Fld@format@code\@empty
3190     \else
3191         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3192     \fi
3193     \ifx\Fld@validate@code\@empty
3194     \else
3195         /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3196     \fi
3197     \ifx\Fld@calculate@code\@empty
3198     \else
3199         /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3200     \fi
3201     \ifx\Fld@onfocus@code\@empty
3202     \else
3203         /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3204     \fi
3205     \ifx\Fld@onblur@code\@empty
3206     \else
3207         /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3208     \fi
3209     \ifx\Fld@onmousedown@code\@empty
3210     \else
3211         /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3212     \fi
3213     \ifx\Fld@onmouseup@code\@empty
3214     \else
3215         /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3216     \fi
3217     \ifx\Fld@onenter@code\@empty
3218     \else
3219         /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3220     \fi
3221     \ifx\Fld@onexit@code\@empty
3222     \else
3223         /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3224     \fi
3225 }
3226 \ExplSyntaxOn
3227 \def\Fld@additionalactions
3228 {%
3229     \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3230     {

```

```

3231     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3232     {/AA<<\Fld@X@additionalactions>>}
3233   }
3234 }
3235 \ExplSyntaxOff
3236 \def\Fld@annotnames{%
3237   /T(\Fld@name)%
3238   \ifx\Fld@altname\relax
3239   \else
3240     /TU(\Fld@altname)%
3241   \fi
3242   \ifx\Fld@mappingname\relax
3243   \else
3244     /TM(\Fld@mappingname)%
3245   \fi
3246 }
3247 \ExplSyntaxOn
3248 \def\PDFForm@Check
3249 {
3250   /Subtype/Widget
3251   ~\Fld@annotflags
3252   ~\Fld@pageobjref
3253   ~\Fld@annotnames
3254   /FT/Btn
3255   \Fld@flags
3256   /Q~\Fld@align
3257   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3258   /AP
3259   <<
3260     /N
3261     <<
3262       /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3263       /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3264     >>
3265   >>
3266   /MK<<
3267     \int_compare:nNnF {\Fld@rotation}={0}
3268     {
3269       /R~\Fld@rotation
3270     }
3271     \tl_if_empty:NF\Fld@bordercolor
3272     {
3273       /BC[\Fld@bordercolor]
3274     }
3275     \tl_if_empty:NF\Fld@bcolor
3276     {
3277       /BG[\Fld@bcolor]
3278     }
3279     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3280   >>
3281   /DA
3282   (
3283     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3284     \tl_if_empty:NF \Fld@color

```

```

3285     {
3286     \c_space_tl \Fld@color
3287     }
3288   )
3289 /H/P
3290 \legacy_if:nTF {Fld@checked}
3291   {
3292     /V/Yes /AS/Yes
3293   }
3294   {
3295     /V/Off /AS/Off
3296   }
3297 \Fld@additionalactions
3298 }
3299 \ExplSyntaxOff
3300 \ExplSyntaxOn
3301 \def\PDFForm@Push
3302   {
3303     /Subtype/Widget
3304     ~\Fld@annotflags
3305     ~\Fld@pageobjref
3306     ~\Fld@annotnames
3307     /FT/Btn
3308     ~\Fld@flags
3309     /H/P
3310     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3311     \bool_if:nT
3312       {
3313         !\int_compare_p:nNn {\Fld@rotation} = {0}
3314         ||
3315         \tl_if_exist_p:N \Fld@bordercolor
3316       }
3317       {
3318         /MK
3319         <<
3320           \int_compare:nNnF {\Fld@rotation} = {0}
3321           {
3322             /R~\Fld@rotation
3323           }
3324           \tl_if_exist:NT \Fld@bordercolor
3325           {
3326             /BC[\Fld@bordercolor]
3327           }
3328         >>
3329       }
3330     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3331     \Fld@additionalactions
3332   }
3333
3334 \ExplSyntaxOff
3335 \def\PDFForm@List{%
3336 /Subtype/Widget%
3337 \Fld@annotflags
3338 \Fld@pageobjref

```

```

3339 \Fld@annotnames
3340 /FT/Ch%
3341 \Fld@flags
3342 /Q \Fld@align
3343 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3344 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3345     \ifx\Fld@bordercolor\relax\else 1\fi
3346     \ifx\Fld@bcolor\relax \else 1\fi
3347     \space
3348 \else
3349     /MK<<%
3350     \ifnum\Fld@rotation=\z@
3351     \else
3352     /R \Fld@rotation
3353     \fi
3354     \ifx\Fld@bordercolor\relax
3355     \else
3356     /BC[\Fld@bordercolor]%
3357     \fi
3358     \ifx\Fld@bcolor\relax
3359     \else
3360     /BG[\Fld@bcolor]%
3361     \fi
3362     >>%
3363 \fi
3364 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3365     \ifx\Fld@color@empty\else\space\Fld@color\fi)%
3366 \Fld@choices
3367 \Fld@additionalactions
3368 }
3369 \ExplSyntaxOn
3370 \def\PDFForm@Radio
3371 {
3372     /Subtype/Widget
3373     ~\Fld@annotflags
3374     ~\Fld@pageobjref
3375     ~\Fld@annotnames
3376     /FT/Btn
3377     \Fld@flags
3378     /H/P
3379     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3380     /MK<<
3381     \ifnum\Fld@rotation=\z@
3382     \else
3383     /R~\Fld@rotation
3384     \fi
3385     \ifx\Fld@bordercolor\relax
3386     \else
3387     /BC[\Fld@bordercolor]%
3388     \fi
3389     \ifx\Fld@bcolor\relax
3390     \else
3391     /BG[\Fld@bcolor]%
3392     \fi

```



```

3393     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3394 >>
3395 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3396     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3397 \ifx\Fld@default\@empty
3398 /V/Off%
3399 /DV/Off%
3400 \else
3401 /V\Fld@default
3402 /DV\Fld@default
3403 \fi
3404 \Fld@additionalactions
3405 }
3406 \ExplSyntaxOff
3407 \ExplSyntaxOn
3408 % Does an appearance dict make sense here?
3409 \def\PDFForm@Text
3410 {
3411     /Subtype/Widget
3412     ~\Fld@annotflags
3413     ~\Fld@pageobjref
3414     ~\Fld@annotnames
3415     /FT/Tx
3416     ~\Fld@flags
3417     /Q~\Fld@align
3418     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3419     \bool_if:nT
3420     {
3421         !\int_compare_p:nNn {\Fld@rotation} = {0}
3422         ||
3423         \tl_if_exist_p:N \Fld@bordercolor
3424         ||
3425         \tl_if_exist_p:N \Fld@bcolor
3426     }
3427     {
3428         /MK
3429         <<
3430         \int_compare:nNnF {\Fld@rotation} = {0}
3431         {
3432             /R~\Fld@rotation
3433         }
3434         \tl_if_exist:NT \Fld@bordercolor
3435         {
3436             /BC[\Fld@bordercolor]
3437         }
3438         \tl_if_exist:NT \Fld@bcolor
3439         {
3440             /BG[\Fld@bcolor]
3441         }
3442         >>
3443     }
3444 /DA
3445 (
3446     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf

```

```

3447     \tl_if_empty:NF {\c_space_tl\Fld@color}
3448   )
3449   /DV(\Hy@escapestring{\Fld@default})
3450   /V(\Hy@escapestring{\Fld@value})
3451   ~\Fld@additionalactions
3452   \int_compare:nNnT { \Fld@maxlen}>{0}
3453   {
3454     /MaxLen~\Fld@maxlen
3455   }
3456 }
3457 \ExplSyntaxOff
3458
3459 \def\PDFForm@Submit{%
3460   /Subtype/Widget%
3461   \Fld@annotflags
3462   \Fld@pageobjref
3463   \Fld@annotnames
3464   /FT/Btn%
3465   \Fld@flags
3466   /H/P%
3467   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3468   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3469     \ifx\Fld@bordercolor\relax\else 1\fi
3470     \space
3471   \else
3472     /MK<<%
3473     \ifnum\Fld@rotation=\z@
3474     \else
3475       /R \Fld@rotation
3476     \fi
3477     \ifx\Fld@bordercolor\relax
3478     \else
3479       /BC[\Fld@bordercolor]%
3480     \fi
3481     >>%
3482   \fi
3483   /A<<%
3484     /S/SubmitForm%
3485     /F<<%
3486     /FS/URL%
3487     /F(\Hy@escapestring{\Form@action})%
3488     >>%
3489     \Fld@submitflags
3490   >>%
3491   \Fld@additionalactions
3492 }
3493 \ExplSyntaxOn
3494   \def\PDFForm@Reset{%
3495     /Subtype/Widget%
3496     \Fld@annotflags
3497     \Fld@pageobjref
3498     \Fld@annotnames
3499     /FT/Btn%
3500     \Fld@flags

```

```

3501         /H/P%
3502         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3503         \ifcase0\ifnum\Fld@rotation=\z@   \else 1\fi
3504             \ifx\Fld@bordercolor\relax\else 1\fi
3505             \space
3506         \else
3507             /MK<<%
3508                 \ifnum\Fld@rotation=\z@
3509                 \else
3510                     /R~\Fld@rotation
3511                 \fi
3512                 \ifx\Fld@bordercolor\relax
3513                 \else
3514                     /BC[\Fld@bordercolor]%
3515                 \fi
3516             >>%
3517         \fi
3518         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3519         /A<</S/ResetForm>>%
3520         \Fld@additionalactions
3521     }%
3522
3523
3524     %these patterns are used in hyperref checks.
3525     %it is unclear if they are really useful and if a backend support is
3526     %needed.
3527     \str_case:VnF \c_sys_backend_str
3528     {
3529         { pdfmode }
3530         {
3531             \def\HyPat@ObjRef
3532             {
3533                 [0-9]*[1-9][0-9]*~0~R
3534             }
3535         }
3536         { dvipdfmx }
3537         {
3538             \def\HyPat@ObjRef
3539             {
3540                 @[\^~]+
3541             }
3542         }
3543         { xdvipdfmx }
3544         {
3545             \def\HyPat@ObjRef
3546             {
3547                 @[\^~]+
3548             }
3549         }
3550     }
3551     { %also set in hyperref sty, so probably not needed.
3552         \def\HyPat@ObjRef/{.+}
3553     }
3554

```

```

3555
3556 \ExplSyntaxOff
3557 % UF: removed Hy@writebookmark
3558 %   \Hy@currentbookmarklevel{0}
3559 %   \Hy@numberline
3560 %   \__hypwritetorep
3561 %   counter{bookmark@seq@number}
3562 % removed \HyPsd@SanitizeForOutFile, not needed
3563 % removed \currentpdfbookmark, defined by bookmark,
3564 % should use \newcommand there
3565 % removed \subpdfbookmark, defined by bookmark,
3566 % should use \newcommand there
3567 % removed \belowpdfbookmark, defined by bookmark,
3568 % should use \newcommand there
3569 % removed \pdfbookmark, defined by bookmark,
3570 % \BOOKMARK
3571 % \@BOOKMARK
3572 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3573 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3574 %% removed \ReadBookmarks / unneeded with bookmark.
3575 %% removed \Hy@OutlineName
3576 %% removed \check@bm@number
3577 %% removed \calc@bm@number
3578
3579 \ifHy@implicit
3580 \else
3581   \expandafter\endinput
3582 \fi
3583 \newlength\Hy@SectionHShift
3584 \def\Hy@SectionAnchorHref#1{%
3585   \ifx\protect\@typeset@protect
3586     \Hy__hypSectionAnchor{#1}%
3587   \fi
3588 }
3589 \DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
3590   \leavevmode
3591   \hbox to 0pt{%
3592     \kern-\Hy@SectionHShift
3593     \Hy@raisedlink{%
3594       \hyper@anchorstart{#1}\hyper@anchorend
3595     }%
3596     \hss
3597   }%
3598 }
3599 \@ifundefined{hyper@nopatch@sectioning}
3600 {
3601   \let\H@old@ssect\@ssect
3602   \def\@ssect#1#2#3#4#5{%
3603     \Hy@MakeCurrentHrefAuto{section*}%
3604     \setlength{\Hy@SectionHShift}{#1}%
3605     \begingroup
3606     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3607     \toks\tw@\expandafter{%
3608       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%

```

```

3609     #5%
3610   }%
3611   \edef\x{\endgroup
3612     \the\toks0{\the\toks\tw}%
3613   }\x
3614 }
3615 \let\H@old@schapter\@schapter
3616 \def\@schapter#1{%
3617   \begingroup
3618     \let\@mkboth\@gobbletwo
3619     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3620     \Hy@raisedlink{%
3621       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3622     }%
3623   \endgroup
3624   \H@old@schapter{#1}%
3625 }
3626 \@ifundefined{@chapter}{}{%
3627   \let\Hy@org@chapter\@chapter
3628   \def\@chapter{%
3629     \def\Hy@next{%
3630       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3631       \Hy@raisedlink{%
3632         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3633       }%
3634     }%
3635     \ifnum\c@secnumdepth>\m@ne
3636       \@ifundefined{if@mainmatter}%
3637       \iftrue{\csname if@mainmatter\endcsname}%
3638       \let\Hy@next\relax
3639     \fi
3640     \fi
3641     \Hy@next
3642     \Hy@org@chapter
3643   }%
3644 }
3645 \let\H@old@part\@part
3646 \begingroup\expandafter\expandafter\expandafter\endgroup
3647 \expandafter\ifx\csname chapter\endcsname\relax
3648   \let\Hy@secnum@part\z@
3649 \else
3650   \let\Hy@secnum@part\m@ne
3651 \fi
3652 \def\@part{%
3653   \ifnum\Hy@secnum@part>\c@secnumdepth
3654     \phantomsection
3655     \fi
3656   \H@old@part
3657 }
3658 \let\H@old@spart\@spart
3659 \def\@spart#1{%
3660   \Hy@MakeCurrentHrefAuto{part*}%
3661   \Hy@raisedlink{%
3662     \hyper@anchorstart{\@currentHref}\hyper@anchorend

```

```

3663 }%
3664 \H@old@spart{#1}%
3665 }
3666 \let\H@old@sect\@sect
3667 \def\@sect#1#2#3#4#5#6[#7]#8{%
3668   \ifnum #2>\c@secnumdepth
3669     \expandafter\@firstoftwo
3670   \else
3671     \expandafter\@secondoftwo
3672   \fi
3673   {%
3674     \Hy@MakeCurrentHrefAuto{section*}%
3675     \setlength{\Hy@SectionHShift}{#3}%
3676     \begingroup
3677     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3678     \toks\tw@\expandafter{%
3679       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3680       #8%
3681     }%
3682     \edef\x{\endgroup
3683       \the\toks@{\the\toks\tw@}%
3684     }\x
3685   }{%
3686     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3687   }%
3688 }
3689 }{}
3690 \expandafter\def\csname Parent-4\endcsname{}
3691 \expandafter\def\csname Parent-3\endcsname{}
3692 \expandafter\def\csname Parent-2\endcsname{}
3693 \expandafter\def\csname Parent-1\endcsname{}
3694 \expandafter\def\csname Parent0\endcsname{}
3695 \expandafter\def\csname Parent1\endcsname{}
3696 \expandafter\def\csname Parent2\endcsname{}
3697 \expandafter\def\csname Parent3\endcsname{}
3698 \expandafter\def\csname Parent4\endcsname{}
3699 %%
3700 %% End of file 'hgeneric-testphase.def'.
3701 </package>

3702 <*colorscheme>
3703 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3704 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3705 % linkcolor=
3706 %,filecolor=
3707 %,urlcolor=
3708 %,menucolor=
3709 %,runcolor=
3710 %,linkbordercolor=
3711 %,filebordercolor=
3712 %,urlbordercolor=
3713 %,menubordercolor=
3714 %,runbordercolor=
3715
3716 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }

```

```

3717 {
3718   linkcolor      = [rgb]{1,0,0}, %red
3719   filecolor      = [rgb]{0,1,1}, %cyan
3720   urlcolor       = [rgb]{1,0,1}, %magenta
3721   menucolor      = [rgb]{1, 0, 0}, %red
3722   runcolor       = [rgb]{0,1,1}, %cyan
3723 %-----
3724   linkbordercolor = [rgb]{1, 0 ,0 },
3725   filebordercolor = [rgb]{0, .5, .5},
3726   urlbordercolor  = [rgb]{0, 1, 1},
3727   menubordercolor = [rgb]{1, 0, 0},
3728   runbordercolor  = [rgb]{0, .7, .7}
3729 }
3730
3731 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3732 {
3733   linkcolor      = [rgb]{0,0.2,0.6},
3734   filecolor      = [rgb]{0.8,0,0.8},
3735   urlcolor       = [rgb]{0.8,0,0.8},
3736   menucolor      = [rgb]{0,0.2,0.6},
3737   runcolor       = [rgb]{0.8,0,0.8},
3738 %----- %-----
3739   linkbordercolor = [rgb]{0,0.2,0.6},
3740   filebordercolor = [rgb]{0.8,0,0.8},
3741   urlbordercolor  = [rgb]{0.8,0,0.8},
3742   menubordercolor = [rgb]{0,0.2,0.6},
3743   runbordercolor  = [rgb]{0.8,0,0.8}
3744 }
3745
3746 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3747 { %two colors: intern/extern
3748   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3749   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3750   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3751   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3752   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3753 %----- %-----
3754   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3755   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3756   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3757   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3758   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3759 }
3760
3761 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3762 { %all darkgray
3763   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3764   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3765   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3766   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3767   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3768 %----- %-----
3769   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3770   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },

```

```

3771     urlbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3772     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3773     runbordercolor = [rgb]{0.4 ,0.4 ,0.4 }
3774 }
3775
3776 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3777 { %dvipsnam.def
3778     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3779     filecolor      = [rgb]{1, 0, 0}, %Red
3780     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3781     menucolor      = [rgb]{1, 0, 0}, %Red
3782     runcolor       = [rgb]{1, 0, 0}, %Red
3783     %----- %-----
3784     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3785     filebordercolor = [rgb]{1, 0, 0}, %Red
3786     urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3787     menubordercolor = [rgb]{1, 0, 0}, %Red
3788     runbordercolor = [rgb]{1, 0, 0} %Red
3789 }
3790
3791 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3792 { %dvipsnam.def
3793     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3794     filecolor      = [rgb]{0, 1, 0}, %Green
3795     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3796     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3797     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3798     %----- %-----
3799     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3800     filebordercolor = [rgb]{0, 1, 0}, %Green
3801     urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3802     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3803     runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3804 }
3805
3806
3807 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3808 {
3809     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3810     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3811     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3812     menucolor      = [rgb]{0.44706, 0.45882, 0},
3813     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3814     %----- %-----
3815     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3816     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3817     urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3818     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3819     runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3820 }
3821
3822 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3823 {
3824     linkcolor      = [rgb]{0,0,0},

```



```
3825   filecolor      = [rgb]{0,0,0},
3826   urlcolor       = [rgb]{0,0,0},
3827   menucolor      = [rgb]{0,0,0},
3828   runcolor       = [rgb]{0,0,0},
3829 %----- %-----
3830   linkbordercolor = [rgb]{0,0,0},
3831   filebordercolor = [rgb]{0,0,0},
3832   urlbordercolor  = [rgb]{0,0,0},
3833   menubordercolor = [rgb]{0,0,0},
3834   runbordercolor  = [rgb]{0,0,0}
3835 }
3836 </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**
- `\#` ..... 280, 804
  - `\$` ..... 279
  - `\%` ..... 805
  - `\-` ..... 2222
  - `\.` ..... 566, 570, 572
  - @curropt commands:
    - `\@curropt:` ..... 2743, 2810
  - `\[` ..... 2222
  - `\]` ... 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, 955, 2745, 2812, 3152
  - `\_` ..... 564, 566, 570, 572
  - `\]` ..... 2222
- A**
- `\A` ..... 564, 2222
  - `\Acrobatmenu` ..... 18, 174
  - `\addcontentsline` ..... 13
  - `\AddToDocumentProperties` . 389, 2217, 2320
  - `\AddToHook` ..... 427, 444, 450
  - `\AddToHookNext` ..... 199
  - `\advance` ..... 2675, 2750, 2794, 2795
  - `allcolors` (hypersetup key) ..... 1083
  - `\author` ..... 2
- B**
- `\b` ..... 568
  - `\begingroup` ..... 210,  
278, 376, 2581, 2710, 2739, 2751,  
2785, 3174, 3605, 3617, 3646, 3676
  - `\belowpdfbookmark` ..... 3567
  - `\bgroup` ..... 275, 376
  - `\BOOKMARK` ..... 3570
  - `bookmarkstype` (hypersetup key) ..... 13
  - bool commands:
    - `\bool_if:NTF` ..... 268,  
297, 356, 724, 748, 768, 788, 799,  
835, 901, 966, 1044, 1054, 1307, 1318
    - `\bool_if:nTF` 916, 1992, 2017, 3311, 3419
    - `\bool_lazy_and:nnTF` ..... 424
    - `\bool_lazy_or:nnTF` ..... 1337, 1386
    - `\bool_new:N` ... 204, 205, 550, 554, 558
    - `\bool_set_true:N` ..... 559
  - `bordercolormodel` (hypersetup key) 13, 1104
  - box commands:
    - `\box_dp:N` ..... 661, 2895, 2946, 2981
    - `\box_ht:N` ..... 660, 2894, 2945, 2980
    - `\box_new:N` ..... 464, 561
    - `\box_set_dp:Nn` ..... 2189
    - `\box_set_ht:Nn` ..... 2196
    - `\box_use:N` .... 1324, 2897, 2955, 2984
    - `\box_use_drop:N` ..... 1329
    - `\box_wd:N` ..... 659, 2893, 2944, 2979
    - `\l_tmpa_box` ..... 2891, 2893, 2894,  
2895, 2897, 2930, 2944, 2945, 2946,  
2955, 2977, 2979, 2980, 2981, 2984
- C**
- `\catcode` ..... 279, 280
  - `\char` ..... 2829, 3026, 3038
  - `\chardef` ..... 159
  - `\cite` ..... 34
  - clist commands:
    - `\clist_item:nn` ..... 2223, 2252
    - `\clist_map_function:nN` ..... 125, 134
    - `\clist_map_inline:nn` ..... 2384
    - `\clist_map_variable:nNn` .. 2742, 2809
  - color commands:
    - `\color_export:nnN` . 41, 435, 1003, 1125
    - `\color_select:n` .....  
..... 1011, 1047, 1328, 2936, 2940
    - `\color_select:nn` ..... 1017
    - `\color_set:nn` ..... 2, 394, 1028
    - `\color_set:nnn` ..... 2, 393, 1034
  - color names:
    - `hyp/annot/file` ..... 547
    - `hyp/annot/link` ..... 547
    - `hyp/annot/menu` ..... 547
    - `hyp/annot/run` ..... 547
    - `hyp/annot/url` ..... 547
  - `colorfile` (hypersetup key) ..... 1083
  - `colorlink` (hypersetup key) ..... 1083
  - `colorlinks` (hypersetup key) ..... 1060
  - `colormenu` (hypersetup key) ..... 1083
  - `colorrun` (hypersetup key) ..... 1083
  - `colorscheme` (hypersetup key) ..... 1, 1495
  - `colorurl` (hypersetup key) ..... 1083
  - cs commands:
    - `\cs_generate_variant:Nn` .....  
..... 156, 157, 158, 391, 463,  
602, 609, 832, 1003, 1020, 1037, 2223
    - `\cs_gset:Npn` ..... 1299
    - `\cs_gset_eq:NN` ..... 410, 421
    - `\cs_if_exist:NTF` ..... 13

<code>\cs_if_exist_p:N</code> . . . . .	425	dim commands:	
<code>\cs_new:Npn</code> . . . . .	118, 633	<code>\dim_to_decimal_in_bp:n</code> . . . . .	12, 162
<code>\cs_new_protected:Npn</code> . . . . .	210,	<code>\do</code> . . . . .	2743, 2810
306, 323, 340, 366, 386, 401, 412,		<code>\documentclass</code> . . . . .	22, 34
457, 578, 582, 586, 590, 603, 605,		<code>\DocumentMetadata</code> . . . . .	
610, 674, 682, 691, 695, 717, 722,		. . .	3, 10, 14, 15, 18, 13, 21, 33, 42, 48
746, 766, 786, 833, 899, 964, 1004,		<code>draft</code> (hypersetup key) . . . . .	1519
1015, 1021, 1032, 1243, 2224, 2306			
<code>\cs_set:Nn</code> . . . . .	216, 258		
<code>\cs_set_eq:NN</code> . . . . .	162,	<b>E</b>	
167, 168, 432, 641, 643, 804, 805, 1553		<code>\edef</code> 2583, 2753, 2789, 2796, 3137, 3611, 3682	
<code>\cs_set_protected:Npn</code> . . . . .		<code>\EdefEscapeName</code> . . . . .	2808, 2813
. . . . .	160, 181, 620, 955	<code>\egroup</code> . . . . .	379
<code>\csname</code> . . . . .	2757, 3144,	<code>\else</code> . . . . .	2580, 2706, 2760,
3155, 3637, 3647, 3690, 3691, 3692,		2764, 2767, 2784, 2792, 3125, 3130,	
3693, 3694, 3695, 3696, 3697, 3698		3147, 3153, 3165, 3169, 3186, 3190,	
<code>\currentpdfbookmark</code> . . . . .	3563	3194, 3198, 3202, 3206, 3210, 3214,	
		3218, 3222, 3239, 3243, 3344, 3345,	
		3346, 3348, 3351, 3355, 3359, 3365,	
		3382, 3386, 3390, 3396, 3400, 3468,	
		3469, 3471, 3474, 3478, 3503, 3504,	
		3506, 3509, 3513, 3580, 3649, 3670	
		<code>\endcsname</code> . . . . .	2771, 3121, 3144,
		3155, 3637, 3647, 3690, 3691, 3692,	
		3693, 3694, 3695, 3696, 3697, 3698	
		<code>\endgroup</code> . . . . .	284, 287,
		377, 379, 2584, 2730, 2753, 2799,	
		2802, 3181, 3611, 3623, 3646, 3682	
		<code>\endinput</code> . . . . .	3581
<b>D</b>		exp commands:	
<code>\d</code> . . . . .	566, 570, 572	<code>\exp_args:Ne</code> . . . . .	842, 2334, 2575, 3229
<code>debug</code> (hypersetup key) . . . . .	1519	<code>\exp_args:Nne</code> . . . . .	2349, 2529
<code>\DeclareRobustCommand</code> . . . . .		<code>\exp_args:Nnno</code> . . . . .	325, 342
. . . . .	263, 291, 314, 331, 350, 379, 3589	<code>\exp_args:Nno</code> . . . . .	308, 368, 2292
<code>\def</code> . . . . .	7, 166, 376, 377, 379,	<code>\exp_args:NNV</code> . . . . .	1977, 2008, 2253
615, 2544, 2548, 2555, 2573, 2602,		<code>\exp_args:NnV</code> . . . . .	676, 685
2615, 2673, 2683, 2698, 2699, 2702,		<code>\exp_args:No</code> . . . . .	
2703, 2716, 2717, 2733, 2734, 2737,		. . . . .	709, 713, 2874, 2917, 2965, 3000
2738, 2776, 2805, 2857, 2871, 2872,		<code>\exp_args:NV</code> . . . . .	657, 664
2884, 2912, 2914, 2915, 2925, 2960,		<code>\exp_not:N</code> . . . . .	955
2962, 2963, 2974, 2993, 2995, 2996,		<code>\exp_not:n</code> . . . . .	
2998, 2999, 3008, 3055, 3122, 3132,		1434, 1468, 1560, 1615, 1636, 1673,	
3134, 3139, 3151, 3162, 3173, 3184,		1693, 1716, 1760, 1793, 1818, 1869,	
3227, 3236, 3248, 3301, 3335, 3370,		1894, 1923, 1972, 2049, 2077, 2104,	
3409, 3459, 3494, 3531, 3538, 3545,		2124, 2370, 2474, 2502, 2512, 2523	
3552, 3584, 3602, 3616, 3628, 3629,		<code>\expandafter</code> . . . . .	281, 2711, 2745,
3652, 3659, 3667, 3690, 3691, 3692,		2754, 2756, 2757, 2812, 3138, 3154,	
3693, 3694, 3695, 3696, 3697, 3698		3581, 3607, 3608, 3646, 3647, 3669,	
<code>\DefaultHeightofCheckBox</code> . . . . .	2999	3671, 3678, 3679, 3690, 3691, 3692,	
<code>\DefaultHeightofChoiceMenu</code> . . . . .	2738	3693, 3694, 3695, 3696, 3697, 3698	
<code>\DefaultHeightofReset</code> . . . . .	2963	<code>\ExplSyntaxOff</code> . . . . .	
<code>\DefaultHeightofSubmit</code> . . . . .	2915	. . . . .	10, 2612, 2665, 2693, 3051, 3119,
<code>\DefaultHeightofText</code> . . . . .	2707		3235, 3299, 3334, 3406, 3457, 3556
<code>\DefaultHeightofTextMultiline</code> . . . . .	2705		
<code>\DefaultOptionsofCheckBox</code> . . . . .	3002		
<code>\DefaultOptionsofPushButton</code> . . . . .	2876		
<code>\DefaultOptionsofReset</code> . . . . .	2967		
<code>\DefaultOptionsofSubmit</code> . . . . .	2919		
<code>\DefaultOptionsofText</code> . . . . .	2712		
<code>\DefaultWidthofCheckBox</code> . . . . .	2998		
<code>\DefaultWidthofChoiceMenu</code> . . . . .	2737		
<code>\DefaultWidthofReset</code> . . . . .	2962		
<code>\DefaultWidthofSubmit</code> . . . . .	2914		
<code>\DefaultWidthofText</code> . . . . .	2702		
<code>destlabel</code> (hypersetup key) . . . . .	13		

<code>\ExplSyntaxOn</code> .....		<code>\hbox_set_end:</code> .....	1320
... 8, 11, 2613, 2670, 2694, 3054,		<code>\hbox_set_to_wd:Nnn</code> .....	2173
3226, 3247, 3300, 3369, 3407, 3493		<code>hidefile</code> (hypersetup key) .....	1471
<code>extension</code> (hypersetup key) .....	13, 1537	<code>hidelink</code> (hypersetup key) .....	1471
<b>F</b>			
<code>\fbox</code> .....	2848, 2936, 2940	<code>hidelinks</code> (hypersetup key) .....	1471
<code>\fi</code> .....	7, 2569,	<code>hidemenu</code> (hypersetup key) .....	1471
2596, 2599, 2608, 2708, 2716, 2717,		<code>hiderun</code> (hypersetup key) .....	1471
2748, 2766, 2769, 2770, 2776, 2779,		<code>hideurl</code> (hypersetup key) .....	1471
2788, 2790, 2797, 2800, 3127, 3149,		<code>hook</code> commands:	
3159, 3161, 3167, 3171, 3178, 3179,		<code>\hook_gput_code:nnn</code> .....	
3188, 3192, 3196, 3200, 3204, 3208,		... 403, 414, 1040, 1050, 1303, 1314	
3212, 3216, 3220, 3224, 3241, 3245,		<code>\hook_new:n</code> .....	392, 577, 673
3344, 3345, 3346, 3353, 3357, 3361,		<code>\hook_new_pair:nn</code> .....	
3363, 3365, 3384, 3388, 3392, 3396,		... 261, 289, 313, 330, 348	
3403, 3468, 3469, 3476, 3480, 3482,		<code>\hook_use:n</code> .....	
3503, 3504, 3511, 3515, 3517, 3582,		. 265, 285, 294, 310, 317, 327, 334,	
3587, 3639, 3640, 3651, 3655, 3672		344, 353, 372, 594, 679, 688, 733, 757	
<code>file</code> (hypersetup key) .....	10, 1568	<code>\href</code> .....	4, 5, 19, 263
<code>file</code> commands:		<code>\hreflaunch</code> .....	5
<code>\file_input:n</code> .....	9	<code>\hrefpdf</code> .....	5, 19, 314, 384
<code>fileborderstyle</code> (hypersetup key) .....	14, 1197	<code>\hrefrun</code> .....	5, 331, 385
<code>filecolor</code> (hypersetup key) .....	1083	<code>\hrefurl</code> .....	5, 19, 291, 383
<code>final</code> (hypersetup key) .....	1519	<code>\hss</code> .....	3596
<code>\fontencoding</code> .....	2825, 3022, 3034	<code>Hy</code> internal commands:	
<code>\fontfamily</code> .....	2824, 3021, 3033	<code>\Hy_hyescapeform</code> ..	3138, 3151, 3157
<code>\fontseries</code> .....	2826, 3023, 3035	<code>\Hy_hypSectionAnchor</code> ...	3586, 3589
<code>\fontshape</code> .....	2827, 3024, 3036	<code>HyField</code> internal commands:	
<code>fp</code> commands:		<code>\HyField_hypAddToFields</code> .	2575, 2602
<code>\fp_eval:n</code> .....	2147, 2175, 2191	<code>hyp</code> commands:	
<b>G</b>			
<code>\gdef</code> .....	281, 282	<code>\l_hyp_annot_colorfile_bool</code> .	7, 548
<code>\GetDocumentProperties</code> .....	3	<code>\l_hyp_annot_colorlink_bool</code> .	7, 548
<code>\global</code> .....	2675, 3117	<code>\l_hyp_annot_colormenu_bool</code> .	7, 548
<code>group</code> commands:		<code>\l_hyp_annot_colorrrun_bool</code> ..	7, 548
<code>\group_begin:</code> .....	266, 295, 318,	<code>\l_hyp_annot_colorurl_bool</code> ..	7, 548
335, 354, 592, 730, 740, 754, 761,		<code>\l_hyp_annot_ocgcolorfile_bool</code> 7,	552
793, 821, 840, 889, 906, 948, 971,		<code>\l_hyp_annot_ocgcolorlink_bool</code> 7,	552
997, 1046, 1310, 1327, 2453, 2823,		<code>\l_hyp_annot_ocgcolormenu_bool</code> 7,	552
2873, 2916, 2964, 2997, 3020, 3032		<code>\l_hyp_annot_ocgcolorrrun_bool</code> 7,	552
<code>\group_end:</code> .....	309, 326,	<code>\l_hyp_annot_ocgcolorurl_bool</code> 7,	552
343, 371, 599, 736, 740, 773, 777,		<code>\l_hyp_current_dest_name_tl</code> ....	
818, 821, 886, 889, 946, 948, 995,		... 672, 731, 755	
997, 1056, 1330, 1333, 2460, 2830,		<code>hyp</code> internal commands:	
2909, 2957, 2990, 3027, 3039, 3049		<code>\g__hyp_AcroForm_CoFields_prop</code> ..	
<b>H</b>			
<code>\hbox</code> .....	376, 3591	... 2539, 2557, 2626, 2628, 2643	
<code>hbox</code> commands:		<code>\g__hyp_AcroForm_Fields_prop</code> ...	
<code>\hbox_overlap_right:n</code> .....	1324	... 2540, 2550, 2621	
<code>\hbox_set:Nn</code> .....	2891, 2930, 2977	<code>\l__hyp_annot_GoTo_bool</code> 724, 748, 768	
<code>\hbox_set:Nw</code> .....	1311	<code>\l__hyp_annot_GoTo_bool_uuuu\l__</code>	
		<code>hyp_annot_URI_bool_uuuu\l__</code>	
		<code>hyp_annot_GoToR_bool_uuuu\l__</code>	
		<code>hyp_annot_Named_bool_uuuu\l__</code>	
		<code>hyp_annot_Launch_bool</code> .....	556
		<code>\l__hyp_annot_GoToR_bool</code> .....	835
		<code>\l__hyp_annot_Launch_bool</code> .....	901

<code>\l__hyp_annot_Named_bool</code> . . . . .	966	<code>\l__hyp_filename_tmpa_tl</code> . . .	470,
<code>\c__hyp_annot_types_seq</code> . . . . .		841, 843, 848, 849, 854, 910, 911, 929	
. . . . .	477, 548, 552, 1083, 1384, 1482	<code>\__hyp_href_pdf_aux:nn</code> . . . . .	320, 323
<code>\l__hyp_annot_URI_bool</code> . . . . .	788	<code>\l__hyp_href_pdf_destination_tl</code> .	
<code>\g__hyp_bordercolormodel_str</code> . . .		. . . . .	207, 218, 257, 325
. . . . .	437, 547, 1108, 1127	<code>\l__hyp_href_pdf_page_tl</code> 208, 226, 866	
<code>\__hyp_check_link_nesting:TF</code> . . .		<code>\__hyp_href_run_aux:nn</code> . . . . .	337, 340
. . . . .	633, 641,	<code>\l__hyp_href_run_parameter_tl</code> . . .	
643, 727, 751, 770, 791, 838, 904, 969		. . . . .	209, 230, 342
<code>\__hyp_citebordercolor_hook_-</code>		<code>\__hyp_href_url_aux:n</code> . . . . .	363, 366
init: . . . . .	399, 412, 421	<code>\__hyp_href_url_aux:nn</code> . . . . .	304, 306
<code>\__hyp_citecolor_hook_init:</code> . . . .		<code>\l__hyp_href_url_encode_bool</code> . . .	
. . . . .	397, 401, 410	. . . . .	204, 215, 253, 268, 297, 356
<code>\__hyp_clist_display:n</code> . 118, 125, 134		<code>\__hyp_href_url_format:</code> . . . . .	
<code>\__hyp_color_select:n</code> . . . . .		. . . . .	210, 216, 258, 369
. . . . .	41, 1004, 1004, 1020	<code>\l__hyp_href_url_ismap_bool</code> . . . .	
<code>\__hyp_color_select_aux:wn</code> . . . . .		. . . . .	205, 229, 799
. . . . .	1004, 1008, 1015	<code>\l__hyp_href_url_protocol_tl</code> . . .	
<code>\__hyp_color_set:nn</code> . . 41, 42, 397,		. . . . .	206, 217, 256, 308, 370
399, 434, 1021, 1021, 1037, 1088, 1124		<code>\__hyp_if_outer_link:</code> . . . . .	627
<code>\__hyp_color_set_aux:nwn</code> . . . . .		<code>\__hyp_if_outer_link:TF</code> . . . . .	643
. . . . .	1021, 1025, 1032	<code>\l__hyp_link_Contents_tl</code> . . . . .	
<code>\c__hyp_colorscheme_daleif_prop</code> 3731		. . . . .	512, 521, 523, 530, 532, 539, 541
<code>\c__hyp_colorscheme_henryford_-</code>		<code>\__hyp_link_goto_begin:nw</code> . . . . .	
prop . . . . .	3822	. . . . .	695, 734, 758
<code>\c__hyp_colorscheme_julian_prop</code> 3746		<code>\__hyp_link_goto_end:</code> . 717, 735, 772	
<code>\c__hyp_colorscheme_phelype_prop</code>		<code>\g__hyp_linknestlevel_int</code> . . . . .	
. . . . .	3807	. . . . .	626, 629, 726, 742, 750, 779,
<code>\c__hyp_colorscheme_szabolcsA_-</code>		790, 823, 837, 891, 903, 949, 968, 998	
prop . . . . .	3776	<code>g__hyp_linknestlevel_int</code> . . . . .	626
<code>\c__hyp_colorscheme_szabolcsB_-</code>		<code>\c__hyp_map_annot_hyp_prop</code> . . . .	477
prop . . . . .	3791	<code>\c__hyp_map_hyp_annot_prop</code> . . . . .	
<code>\c__hyp_colorscheme_tivv_prop</code> . 3761		. . . . .	477, 1038,
<code>\l__hyp_dest_box</code> . . . . .	33,	1111, 1151, 1178, 1186, 1197, 1224,	
561, 659, 660, 661, 2173, 2189, 2196		1232, 1301, 1409, 1445, 1455, 1568	
<code>\l__hyp_dest_name_tmpa_tl</code> . . 470,		<code>\__hyp_ocg_init:</code> <u>1243</u> , 1243, 1299, 1309	
698, 699, 705, 709, 711, 714, 858, 872		<code>\l__hyp_optlang_regex</code> 2221, 2222, 2254	
<code>\l__hyp_dest_pdfremotestartview_-</code>		<code>l__hyp_page/Trans</code> . . . . .	575
tl . . . . .	496, 867, 1980, 1984	<code>\__hyp_PageLabels_gpush:</code> 610, 610, 623	
<code>\g__hyp_dest_pdfstartpage_tl</code> . . .		<code>\l__hyp_para_tmpa_tl</code> 473, 915, 918, 929	
. . . 496, 1991, 1993, 2000, 2018, 2025		<code>\l__hyp_para_tmpa_tl\l__hyp_-</code>	
<code>\g__hyp_dest_pdfstartview_tl</code> . . .		text_tmpa_str\l__hyp_text_-	
. . . 496, 1993, 2000, 2011, 2015, 2018, 2025		tmpa_str . . . . .	470
<code>\l__hyp_dest_pdfview_tl</code> . . . . .		<code>\__hyp_property_record:nn</code> . . . . .	
. . . . .	546, 677, 686,	. . . . .	456, 457, 463, 2679
2142, 2145, 2152, 2155, 2156, 2157,		<code>\__hyp_secondoftwewithopt:wnn</code> . . .	
2158, 2159, 2160, 2165, 2169, 2205		. . . . .	382, 383, 384, 385
<code>\c__hyp_dest_startview_regex</code> . . .		<code>\__hyp_setup_info_date_key:nn</code> . . .	
. . . . .	562, 1978, 2009	. . . . .	2306, 2339, 2340
<code>\c__hyp_dest_undefined_tl</code> . . . . .		<code>\__hyp_setup_info_key:nn</code> . . 2224,	
. . . . .	476, 704, 705	2299, 2300, 2301, 2302, 2304, 2305	
<code>\__hyp_destination:nn</code> . . . . .		<code>\__hyp_store_metadata:nn</code> . . . . .	
. . . . .	33, 646, 646, 677, 686	. . . . .	386, 391, 1587, 2231,
		2274, 2319, 2334, 2343, 2359, 2427	

\_hyp\_text\_cleanup:N	582, 582, 596	bookmarkstype	13
\l\_hyp\_text\_enc\_dest\_print\_tl	499, 857	bordercolormodel	13, 1104
\l\_hyp\_text\_enc\_dest\_tl	33, 499, 653, 710	colorfile	1083
\l\_hyp\_text\_enc\_file\_print\_tl	499, 909	colorlink	1083
\l\_hyp\_text\_enc\_info\_print\_tl	499, 520, 529, 538, 607	colorlinks	1060
\l\_hyp\_text\_enc\_para\_print\_tl	499, 914	colormenu	1083
\l\_hyp\_text\_enc\_uri\_print\_tl	270, 273, 299, 302, 358, 361, 499, 796, 1579	colorrund	1083
\_hyp\_text\_pdfstring:nnN	518, 527, 536, 590, 590, 602, 604, 607, 651, 708, 794, 855, 907, 912, 1579	colorscheme	1, 1495
\_hyp\_text\_pdfstring\_info:nN	605, 605, 609, 2257, 2260, 2280, 2289	colorurl	1083
\_hyp\_text\_purify:nN	578, 578, 595	debug	1519
\_hyp\_text\_string\_from\_unicode:nN	586, 586, 597	destlabel	13
\g\_hyp\_text\_tmpa\_str	475, 598, 600	draft	1519
\l\_hyp\_text\_tmpa\_str	474, 595, 596, 597, 598	extension	13, 1537
\l\_hyp\_tmpa\_box	464, 1311, 1324, 1329	file	10, 1568
\l\_hyp\_tmpa\_int	464	fileborderstyle	14, 1197
\l\_hyp\_tmpa\_seq	464, 1978, 1980, 2009, 2011, 2132, 2133, 2137, 2139, 2163, 2170, 2171, 2172, 2181, 2183, 2193, 2198, 2254, 2255, 2260, 2630, 2632, 2638	filecolor	1083
\l\_hyp\_tmpa\_str	464, 2257, 2260, 2262, 2264, 2280, 2281, 2283, 2289, 2290, 2294	final	1519
\l\_hyp\_tmpa\_tl	464, 654, 658, 666, 1128, 1132, 1579, 1580, 1585, 1976, 1978, 2007, 2009, 2139, 2140, 2147, 2230, 2231, 2232, 2252, 2257, 2273, 2274, 2280	hidefile	1471
\l\_hyp\_tmpb\_tl	464, 2252, 2254	hidelink	1471
\l\_hyp\_uri\_tmpa\_tl	470, 797, 798	hidelinks	1471
hyp/anchor	672	hidemenu	1471
hyp/annot/file (color name)	547	hiderun	1471
hyp/annot/link (color name)	547	hideurl	1471
hyp/annot/menu (color name)	547	hypertextnames	1537
hyp/annot/run (color name)	547	link	10, 1568
hyp/annot/url (color name)	547	linkborder	14
hyp/text/pdfstring	577	linkborderstyle	14, 1197
\hypercalcbp	12, 18, 162	linkcolor	1083
\HyperDestNameFilter	13, 652, 709	linkfileprefix	1537
\hypersetup	1, 2, 5, 9, 10, 13, 19, 69, 102, 181	linktoc	1537
\hypersetup keys:		linktocpage	1537
allcolors	1083	localanchorname	1537
		menu	10, 1568
		menuborder	14
		menuborderstyle	14, 1197
		menucolor	1083
		naturalnames	1537
		nested-links	10
		nesting	14
		ocgcolorfile	1337
		ocgcolorlink	1337
		ocgcolorlinks	1337
		ocgcolormenu	1337
		ocgcolorrund	1337
		ocgcolorurl	1337
		pageanchor	1537
		pdfauthor	2221
		pdfborder	14
		pdfborderstyle	14, 1197
		pdfcreationdate	14, 2306
		pdfcreator	2221
		pdfencoding	1509
		pdfinfo	2373
		pdfkeywords	2221

pdflang	14, 2210	iow commands:	
pdflinkmargin	14	\iow_newline:	3070, 3076, 3089
pdfmetadate	14, 2306		
pdfmoddate	14, 2306	<b>K</b>	
pdfproducer	2221	\kern	3592
pdfremotestartview	12	keys commands:	
pdfstartview	12	\keys_define:nn	187, 195, 213, 251, 395, 637, 1060, 1085, 1092, 1104, 1113, 1138, 1153, 1172, 1199, 1218, 1341, 1355, 1367, 1390, 1396, 1411, 1440, 1471, 1484, 1495, 1509, 1519, 1524, 1537, 1549, 1570, 1575, 2128, 2210, 2226, 2269, 2308, 2323, 2341, 2345, 2373, 2425, 2431, 2444, 2464
pdfsubject	2221	\l_keys_key_str	192, 2294, 2529
pdftitle	2221	\keys_set:nn	184, 221, 240, 267, 296, 319, 336, 355, 405, 416, 1501, 1508, 2350, 2377, 2380, 2381, 2382, 2383, 2454, 2529
pdftrapped	2345	\keys_set_known:nn	2532
pdfversion	1509	\kvsetkeys	183, 2617
pdfview	12, 2128		
plainpages	1537	<b>L</b>	
run	10, 1568	\label	9, 13
runborder	14	\LayoutCheckField	3010
runborderstyle	14, 1197	\LayoutChoiceField	2780
runcolor	1083	\LayoutPushButtonField	2886, 2903
unicode	1509	\LayoutTextField	2718
url	10, 1568	\leavevmode	2719, 2816, 3590
urlborder	14	legacy commands:	
urlborderstyle	14, 1197	\legacy_if:nTF	169, 442, 452, 618, 2655, 2882, 2923, 2973, 3006, 3290
urlcolor	1083	\let	383, 384, 385, 744, 813, 825, 882, 893, 942, 951, 1000, 2542, 2543, 2570, 2666, 2667, 2668, 2700, 2701, 2735, 2736, 2778, 3117, 3124, 3126, 3131, 3148, 3182, 3601, 3615, 3618, 3627, 3638, 3645, 3648, 3650, 3658, 3666
verbose	1519	link (hypersetup key)	10, 1568
hypertextnames (hypersetup key)	1537	linkborder (hypersetup key)	14
hypListbox internal commands:		linkborderstyle (hypersetup key)	14, 1197
\_hypListbox	2798, 2857	linkcolor (hypersetup key)	1083
hypRadio internal commands:		linkfileprefix (hypersetup key)	1537
\_hypRadio	2783, 2805	linktoc (hypersetup key)	1537
hypwritetorep internal commands:		linktocpage (hypersetup key)	1537
\_hypwritetorep	3560	localanchorname (hypersetup key)	1537
		\long	7
<b>I</b>			
\ifcase	3344, 3468, 3503	<b>M</b>	
\ifcsname	3121	\MakeButtonField	2891, 2906, 2930, 2977, 2988
\ifdim	2748, 2787, 2788	\MakeChoiceField	2866
\ifnum	3175, 3344, 3350, 3381, 3468, 3473, 3503, 3508, 3635, 3653, 3668	\MakeFieldObject	2533
\iftrue	3637		
\ifx	2579, 2717, 2777, 3152, 3164, 3168, 3185, 3189, 3193, 3197, 3201, 3205, 3209, 3213, 3217, 3221, 3238, 3242, 3345, 3346, 3354, 3358, 3365, 3385, 3389, 3396, 3397, 3469, 3477, 3504, 3512, 3585, 3647		
\immediate	2563, 2566		
int commands:			
\int_compare:nNnTF	629, 2137, 2163, 2849, 3267, 3320, 3430, 3452		
\int_compare_p:nNn	3313, 3421		
\int_eval:n	865		
\int_gdecr:N	742, 779, 823, 891, 949, 998		
\int_gincr:N	726, 750, 790, 837, 903, 968		
\int_max:nn	866		
\int_new:N	468, 626		

<code>\MakeRadioField</code> .....	2848	<code>\paperwidth</code> .....	3
<code>\MakeTextField</code> .....	2727	<code>\PassOptionsToPackage</code> ...	454, 1529, 1534
<code>\mbox</code> .....	1321	pdf commands:	
menu (hypersetup key) .....	10, 1568	<code>\pdf_bdcobject:nn</code> .....	1323, 1326
menuborder (hypersetup key) .....	14	<code>\pdf_destination:nn</code> ....	27, 156, 665
menuborderstyle (hypersetup key) .....	14, 1197	<code>\pdf_destination:nnnn</code> .....	658
menucolor (hypersetup key) .....	1083	<code>\pdf_emc:</code> .....	1325, 1331
mode commands:		<code>\pdf_link_user:nnn</code> .....	2863
<code>\mode_if_horizontal:TF</code> .....	648, 670	<code>\pdf_name_from_unicode_e:n</code> .....	246, 845, 976
<code>\mode_leave_vertical:</code> ..	264, 293,	<code>\pdf_object_if_exist:nTF</code> .....	843
	316, 333, 352, 697, 807, 874, 931,	<code>\pdf_object_new:n</code> .....	1245,
	974, 2860, 2888, 2905, 2927, 2969, 3012	1246, 1247, 1248, 3057, 3058, 3059	
msg commands:		<code>\pdf_object_ref:n</code> .....	157, 854,
<code>\msg_error:nn</code> .....	426, 2902, 2987	1251, 1253, 1275, 1276, 1279, 1283,	
<code>\msg_info:nnn</code> .....	2238, 2243	1288, 1293, 1298, 2648, 2650, 3115	
<code>\msg_line_context:</code> .....	83	<code>\pdf_object_ref_last:</code> ....	876, 2459
<code>\g_msg_module_name_prop</code> .....	12	<code>\pdf_object_unnamed_write:nn</code> ...	875, 2455
<code>\msg_new:nnn</code> 51, 58, 63, 67, 71, 78,		<code>\pdf_object_write:nnn</code> ....	1249,
85, 92, 99, 106, 112, 119, 128, 138, 149		1255, 1265, 1277, 3060, 3102, 3109	
<code>\msg_new:nnnn</code> .....	15, 26, 37	<code>\pdf_pageobject_ref:n</code> .....	158, 2000, 2025, 2687
<code>\msg_warning:nn</code> .....	171, 1515	<code>\pdf_string_from_unicode:nnN</code> ..	588
<code>\msg_warning:nnn</code> .....	180, 701, 992	<code>\pdf_version:</code> .....	1661, 1780, 1809, 1843, 1858,
<code>\msg_warning:nnnn</code> .....		1884, 1909, 1944, 1960, 2065, 2092	
..... 1359, 1400, 1613, 1657,		<code>\pdf_version_compare:NnTF</code> .....	
1691, 1714, 1776, 1805, 1839, 1854,		1651, 1742, 1764, 1800, 1833, 1848,	
1867, 1880, 1905, 1940, 1956, 1983,		1874, 1899, 1927, 1950, 2055, 2082	
2014, 2047, 2061, 2088, 2122, 2204		<code>\pdf_version_compare_p:Nn</code> .....	920, 1338, 1387
<code>\msg_warning:nnnnn</code> .....	191,	<code>\pdf_version_major:</code> .....	168, 1339, 1362, 1388, 1404
1431, 1465, 1557, 1633, 1670, 1757,		<code>\pdf_version_minor:</code> ..	167, 1362, 1404
1790, 1815, 1891, 1918, 1969, 2074,		pdfannot commands:	
2101, 2367, 2471, 2490, 2509, 2520		<code>\pdfannot_box:nnnn</code> .....	
		.. 2722, 2833, 2892, 2943, 2978, 3042	
		<code>\pdfannot_box_ref_last:</code> ..	2577, 2593
		<code>\pdfannot_dict_put:nnn</code> .....	522,
		531, 540, 808, 876, 932, 977, 1129,	
		1164, 1188, 1210, 1234, 1416, 1447	
		<code>\pdfannot_dict_remove:nn</code> ..	1119,
		1159, 1180, 1205, 1226, 1424, 1457	
		<code>\pdfannot_link:nnn</code> .	809, 877, 933, 978
		<code>\pdfannot_link_goto_begin:nw</code> ..	714
		<code>\pdfannot_link_goto_end:</code> .....	719
		<code>\pdfannot_link_margin:n</code> .....	8, 1695
		<code>\c_pdfannot_link_types_seq</code> ....	556
		pdfauthor (hypersetup key) .....	2221
		<code>\pdfbookmark</code> .....	3569
		pdfborder (hypersetup key) .....	14
		pdfborderstyle (hypersetup key) .	14, 1197
		pdfcreationdate (hypersetup key)	14, 2306
<b>N</b>			
naturalnames (hypersetup key) .....	1537		
nested-links (hypersetup key) .....	10		
nesting (hypersetup key) .....	14		
<code>\newcommand</code> .....	165, 3564, 3566, 3568		
<code>\newcount</code> .....	2671, 2856		
<code>\NewDocumentCommand</code> .....	2533		
<code>\NewExpandableDocumentCommand</code> ....	382		
<code>\newlength</code> .....	3583		
<code>\noexpand</code> ...	2754, 2755, 2756, 2757, 3135		
<code>\nolinkurl</code> .....	4		
<b>O</b>			
ocgcolorfile (hypersetup key) .....	1337		
ocgcolorlink (hypersetup key) .....	1337		
ocgcolorlinks (hypersetup key) ....	1337		
ocgcolormenu (hypersetup key) .....	1337		
ocgcolorrun (hypersetup key) .....	1337		
ocgcolorurl (hypersetup key) .....	1337		
<b>P</b>			
pageanchor (hypersetup key) .....	1537		



pdfcreator (hypersetup key) . . . . .	<a href="#">2221</a>	\pdfstringdef . . . . .	<a href="#">8</a> , <a href="#">9</a> , <a href="#">23</a> , <a href="#">29</a> , <a href="#">3180</a>
\pdfdest . . . . .	<a href="#">11</a>	\pdfstringdefDisableCommands . . . . .	
pdffdict commands:			<a href="#">383</a> , <a href="#">384</a> , <a href="#">385</a>
\pdffdict_new:n . . . . .	<a href="#">575</a> , <a href="#">782</a> , <a href="#">828</a> , <a href="#">895</a> , <a href="#">960</a>	pdfsubject (hypersetup key) . . . . .	<a href="#">2221</a>
\pdffdict_put:nnn . . . . .	. . . . . <a href="#">238</a> , <a href="#">245</a> , <a href="#">576</a> , <a href="#">783</a> , <a href="#">784</a> , <a href="#">798</a> , <a href="#">801</a> , <a href="#">829</a> , <a href="#">830</a> , <a href="#">845</a> , <a href="#">851</a> , <a href="#">862</a> , <a href="#">872</a> , <a href="#">896</a> , <a href="#">897</a> , <a href="#">911</a> , <a href="#">926</a> , <a href="#">961</a> , <a href="#">962</a> , <a href="#">975</a> , <a href="#">1721</a> , <a href="#">1722</a> , <a href="#">1726</a> , <a href="#">1727</a> , <a href="#">2468</a> , <a href="#">2478</a> , <a href="#">2482</a> , <a href="#">2485</a> , <a href="#">2487</a> , <a href="#">2506</a> , <a href="#">2515</a> , <a href="#">2517</a>	pdftitle (hypersetup key) . . . . .	<a href="#">2221</a>
\pdffdict_remove:nn . . . . .	<a href="#">235</a> , <a href="#">923</a> , <a href="#">1731</a> , <a href="#">1732</a>	pdftrapped (hypersetup key) . . . . .	<a href="#">2345</a>
\pdffdict_use:n . . . . .	. . . . . <a href="#">808</a> , <a href="#">875</a> , <a href="#">932</a> , <a href="#">938</a> , <a href="#">977</a> , <a href="#">982</a> , <a href="#">2457</a>	pdfversion (hypersetup key) . . . . .	<a href="#">1509</a>
pdfencoding (hypersetup key) . . . . .	<a href="#">1509</a>	pdfview (hypersetup key) . . . . .	<a href="#">12</a> , <a href="#">2128</a>
\pdfescapestring . . . . .	<a href="#">3124</a>	pdfxform commands:	
pdffile commands:		\pdfxform_if_exist:nTF . . . . .	. . . . . <a href="#">2819</a> , <a href="#">2931</a> , <a href="#">3015</a>
\pdffile_embed_file:nnn . . . . .	<a href="#">832</a> , <a href="#">846</a>	\pdfxform_new:nnn . . . . .	. . . . . <a href="#">2535</a> , <a href="#">2821</a> , <a href="#">2934</a> , <a href="#">2938</a> , <a href="#">3017</a> , <a href="#">3029</a>
PDFForm internal commands:		\pdfxform_ref:n . . . . .	. . . . . <a href="#">2843</a> , <a href="#">2844</a> , <a href="#">2950</a> , <a href="#">2951</a> , <a href="#">3262</a> , <a href="#">3263</a>
\PDFForm_hypName . . . . .	. . . . . <a href="#">3163</a> , <a href="#">3166</a> , <a href="#">3170</a> , <a href="#">3173</a>	\phantom . . . . .	<a href="#">3038</a>
pdfinfo (hypersetup key) . . . . .	<a href="#">2373</a>	\phantomsection . . . . .	<a href="#">13</a> , <a href="#">19</a> , <a href="#">3654</a>
pdfkeywords (hypersetup key) . . . . .	<a href="#">2221</a>	plainpages (hypersetup key) . . . . .	<a href="#">1537</a>
pdflang (hypersetup key) . . . . .	<a href="#">14</a> , <a href="#">2210</a>	prg commands:	
pdflinkmargin (hypersetup key) . . . . .	<a href="#">14</a>	\prg_do_nothing: . . . . .	<a href="#">410</a> , <a href="#">421</a>
pdfmanagement commands:		\prg_generate_conditional_- variant:Nnn . . . . .	<a href="#">2682</a>
\pdfmanagement_add:nn . . . . .	<a href="#">1704</a>	\prg_new_conditional:Npnn . . . . .	<a href="#">627</a>
\pdfmanagement_add:nnn . . . . .	. . . . . <a href="#">612</a> , <a href="#">1275</a> , <a href="#">1276</a> , <a href="#">1298</a> , <a href="#">1585</a> , <a href="#">1605</a> , <a href="#">1621</a> , <a href="#">1625</a> , <a href="#">1645</a> , <a href="#">1653</a> , <a href="#">1682</a> , <a href="#">1737</a> , <a href="#">1747</a> , <a href="#">1771</a> , <a href="#">1785</a> , <a href="#">1797</a> , <a href="#">1802</a> , <a href="#">1827</a> , <a href="#">1835</a> , <a href="#">1850</a> , <a href="#">1876</a> , <a href="#">1901</a> , <a href="#">1935</a> , <a href="#">1952</a> , <a href="#">1998</a> , <a href="#">2023</a> , <a href="#">2037</a> , <a href="#">2057</a> , <a href="#">2084</a> , <a href="#">2113</a> , <a href="#">2216</a> , <a href="#">2239</a> , <a href="#">2244</a> , <a href="#">2264</a> , <a href="#">2283</a> , <a href="#">2293</a> , <a href="#">2317</a> , <a href="#">2332</a> , <a href="#">2354</a> , <a href="#">2440</a> , <a href="#">2459</a> , <a href="#">2623</a> , <a href="#">2640</a> , <a href="#">2647</a> , <a href="#">2649</a> , <a href="#">2651</a> , <a href="#">2657</a>	\prg_return_false: . . . . .	<a href="#">630</a>
\pdfmanagement_if_active_p: . . . . .	<a href="#">425</a>	\prg_return_true: . . . . .	<a href="#">631</a>
\pdfmanagement_remove:nn . . . . .	. . . . . <a href="#">1582</a> , <a href="#">1601</a> , <a href="#">1609</a> , <a href="#">1629</a> , <a href="#">1641</a> , <a href="#">1666</a> , <a href="#">1678</a> , <a href="#">1686</a> , <a href="#">1700</a> , <a href="#">1709</a> , <a href="#">1753</a> , <a href="#">1768</a> , <a href="#">1787</a> , <a href="#">1824</a> , <a href="#">1863</a> , <a href="#">1888</a> , <a href="#">1914</a> , <a href="#">1931</a> , <a href="#">1965</a> , <a href="#">1995</a> , <a href="#">2020</a> , <a href="#">2033</a> , <a href="#">2042</a> , <a href="#">2070</a> , <a href="#">2097</a> , <a href="#">2109</a> , <a href="#">2118</a> , <a href="#">2248</a> , <a href="#">2277</a> , <a href="#">2314</a> , <a href="#">2329</a> , <a href="#">2437</a> , <a href="#">2450</a> , <a href="#">2661</a>	prop commands:	
\pdfmanagement_show:n . . . . .	<a href="#">2624</a>	\prop_const_from_keyval:Nn . . . . .	. . . . . <a href="#">479</a> , <a href="#">487</a> , <a href="#">3716</a> , <a href="#">3731</a> , <a href="#">3746</a> , <a href="#">3761</a> , <a href="#">3776</a> , <a href="#">3791</a> , <a href="#">3807</a> , <a href="#">3822</a>
pdfmeta commands:		\prop_gput:Nnn . . . . .	<a href="#">12</a> , <a href="#">2550</a> , <a href="#">2557</a>
\pdfmeta_standard_verify:nnTF . . . . .	. . . . . <a href="#">972</a> , <a href="#">2879</a> , <a href="#">2970</a>	\prop_if_empty:NTF . . . . .	<a href="#">2626</a>
\pdfmeta_standard_verify:nTF . . . . .	. . . . . <a href="#">2653</a> , <a href="#">3231</a>	\prop_item:Nn . . . . .	<a href="#">2643</a>
pdfmetadate (hypersetup key) . . . . .	<a href="#">14</a> , <a href="#">2306</a>	\prop_map_inline:Nn . . . . .	. . . . . <a href="#">1038</a> , <a href="#">1111</a> , <a href="#">1151</a> , <a href="#">1178</a> , <a href="#">1186</a> , <a href="#">1197</a> , <a href="#">1224</a> , <a href="#">1232</a> , <a href="#">1301</a> , <a href="#">1409</a> , <a href="#">1445</a> , <a href="#">1455</a> , <a href="#">1499</a> , <a href="#">1568</a> , <a href="#">2621</a> , <a href="#">2628</a>
pdfmoddate (hypersetup key) . . . . .	<a href="#">14</a> , <a href="#">2306</a>	\prop_new:N . . . . .	<a href="#">2539</a> , <a href="#">2540</a>
pdfproducer (hypersetup key) . . . . .	<a href="#">2221</a>	property commands:	
pdfremotestartview (hypersetup key) . . . . .	<a href="#">12</a>	\property_if_recorded:nn . . . . .	<a href="#">2682</a>
pdfstartview (hypersetup key) . . . . .	<a href="#">12</a>	\property_if_recorded:nnTF . . . . .	<a href="#">2685</a>
		\property_record:nn . . . . .	<a href="#">460</a>
		\property_ref:nn . . . . .	<a href="#">2689</a>
		\property_ref_undefined_warn:nn . . . . .	<a href="#">2680</a>
		\protect . . . . .	<a href="#">744</a> , <a href="#">813</a> , <a href="#">825</a> , <a href="#">882</a> , <a href="#">893</a> , <a href="#">942</a> , <a href="#">951</a> , <a href="#">1000</a> , <a href="#">3585</a>
		\providecommand . . . . .	. . . . . <a href="#">163</a> , <a href="#">164</a> , <a href="#">2564</a> , <a href="#">2567</a> , <a href="#">2732</a> , <a href="#">3120</a>
		\ProvidesFile . . . . .	<a href="#">3</a>
			<b>R</b>
		\ReadBookmarks . . . . .	<a href="#">3574</a>
		\refstepcounter . . . . .	<a href="#">12</a>
		regex commands:	
		\regex_const:Nn . . . . .	<a href="#">562</a>

<code>\regex_extract_once:NnN</code> . . . . .	2254	<code>\c_percent_str</code> . . . . .	805
<code>\regex_extract_once:NnNTF</code> . . . . .	1978, 2009	<code>\str_case:nnTF</code> . . . . .	2234, 3527
<code>\regex_new:N</code> . . . . .	2221	<code>\str_case_e:nnTF</code> . . . . .	2133
<code>\regex_set:Nn</code> . . . . .	2222	<code>\str_compare:nNnTF</code> . . . . .	2634
<code>\relax</code> . . . . .	744, 813, 825, 882, 893, 942, 951, 1000, 2735, 2736, 2777, 3117, 3164, 3168, 3238, 3242, 3345, 3346, 3354, 3358, 3385, 3389, 3469, 3477, 3504, 3512, 3638, 3647	<code>\str_gset:Nn</code> . . . . .	1108
<code>\RemoveFromHook</code> . . . . .	1591	<code>\str_gset_eq:NN</code> . . . . .	598
<code>\RenewDocumentCommand</code> . . . . .	174	<code>\str_head:n</code> . . . . .	2356, 2361
<code>\RequirePackage</code> . . . . .	6, 32, 446, 3572	<code>\str_if_eq:nnTF</code> . . . . .	655, 2262, 2281, 2290, 3229
<code>run</code> (hypersetup key) . . . . .	10, 1568	<code>\str_if_eq_p:nn</code> . . . . .	918, 1339, 1388
<code>runborder</code> (hypersetup key) . . . . .	14	<code>\str_lowercase:n</code> . . . . .	2133, 2274, 2334, 2357, 2362
<code>runborderstyle</code> (hypersetup key) . . . . .	14, 1197	<code>\str_new:N</code> . . . . .	469, 474, 475, 547
<code>runcolor</code> (hypersetup key) . . . . .	1083	<code>\str_set:Nn</code> . . . . .	580
<b>S</b>		<code>\str_set_convert:Nnnn</code> . . . . .	8
<code>\selectfont</code> . . . . .	2828, 3025, 3037	<code>\str_set_eq:NN</code> . . . . .	600
seq commands:		<code>\str_tail:n</code> . . . . .	2357, 2362
<code>\seq_const_from_clist:Nn</code> . . . . .	477	<code>\str_uppercase:n</code> . . . . .	2350, 2356, 2361
<code>\seq_count:N</code> . . . . .	2137, 2163	<code>\string</code> . . . . .	2564, 2567, 2588, 2606, 3140, 3144, 3155
<code>\seq_get_right:NN</code> . . . . .	2139	<code>\subpdfbookmark</code> . . . . .	3565
<code>\seq_if_empty:NTF</code> . . . . .	2255	sys commands:	
<code>\seq_item:Nn</code> . . . . .	1980, 2011, 2133, 2181, 2183, 2193, 2198, 2260	<code>\c_sys_backend_str</code> . . . . .	3527
<code>\seq_map_inline:Nn</code> . . . . .	548, 552, 556, 1083, 1384, 1482, 2638	<b>T</b>	
<code>\seq_new:N</code> . . . . .	467	T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:	
<code>\seq_put_right:Nn</code> . . . . .	2170, 2171, 2172, 2630	<code>\@BOOKMARK</code> . . . . .	3571
<code>\seq_set_split:Nnn</code> . . . . .	2132	<code>\@CheckBox</code> . . . . .	2993
<code>\seq_sort:Nn</code> . . . . .	2632	<code>\@ChoiceMenu</code> . . . . .	2733
<code>\setbox</code> . . . . .	376	<code>\@Form</code> . . . . .	2615
<code>\setlength</code> . . . . .	3604, 3675	<code>\@PushButton</code> . . . . .	2871
<code>\setpdfinkmargin</code> . . . . .	8	<code>\@Reset</code> . . . . .	2960
<code>\settowidth</code> . . . . .	2747	<code>\@Submit</code> . . . . .	2912
<code>\show</code> . . . . .	2807	<code>\@TextField</code> . . . . .	2698
socket commands:		<code>\@bookmarksopenlevel</code> . . . . .	1596
<code>\socket_assign_plug:nn</code> . . . . .	543, 544, 545	<code>\@bsphack</code> . . . . .	459
<code>\socket_new:nn</code> . . . . .	513, 514, 515	<code>\@chapter</code> . . . . .	3627, 3628
<code>\socket_new_plug:nnn</code> . . . . .	516, 525, 534	<code>\@currDisplay</code> . . . . .	2747, 2815
<code>\socket_use:nn</code> . . . . .	803, 859	<code>\@currValue</code> . . . . .	2813, 2843
<code>\socket_use:nnn</code> . . . . .	732, 756	<code>\@currentHref</code> . . . . .	3608, 3621, 3632, 3662, 3679
sort commands:		<code>\@curropt</code> . . . . .	2732, 2742, 2745, 2809, 2812
<code>\sort_return_same:</code> . . . . .	2636	<code>\@empty</code> . . . . .	432, 2542, 2543, 2579, 2666, 2667, 2668, 2700, 2701, 2717, 3185, 3189, 3193, 3197, 3201, 3205, 3209, 3213, 3217, 3221, 3365, 3396, 3397
<code>\sort_return_swapped:</code> . . . . .	2635	<code>\@endForm</code> . . . . .	2666
<code>\space</code> . . . . .	2815, 3347, 3364, 3365, 3395, 3396, 3470, 3502, 3505	<code>\@esphack</code> . . . . .	461
<code>\spacefactor</code> . . . . .	648, 670	<code>\@firstofone</code> . . . . .	2570, 3126, 3131, 3148
<code>\stockwidth</code> . . . . .	3	<code>\@firstoftwo</code> . . . . .	3669
str commands:		<code>\@for</code> . . . . .	2743, 2810
<code>\c_colon_str</code> . . . . .	955	<code>\@gobbletwo</code> . . . . .	3618
<code>\c_hash_str</code> . . . . .	804	<code>\@hyper@launch</code> . . . . .	40, 955
		<code>\@ifnextchar</code> . . . . .	275

<code>\@ifundefined</code>	3140, 3599, 3626, 3636	<code>\Fld@flags</code>	3255, 3308, 3341, 3377, 3416, 3465, 3500
<code>\@mainaux</code>	2563, 2566, 2586, 2605	<code>\Fld@format@code</code>	3189, 3191
<code>\@mkboth</code>	3618	<code>\Fld@height</code>	2703, 2724, 2727, 2738, 2796, 2835, 2848, 2866, 2915, 2963, 2999, 3044
<code>\@ne</code>	2675	<code>\Fld@keystroke@code</code>	3185, 3187
<code>\@nil</code>	3138, 3151, 3157	<code>\Fld@listcount</code>	2806, 2814, 2849, 2856
<code>\@part</code>	3645, 3652	<code>\Fld@mappingname</code>	3168, 3170, 3242, 3244
<code>\@pdfauthor</code>	23	<code>\Fld@maxlen</code>	3452, 3454
<code>\@pdfborder</code>	163	<code>\Fld@menulength</code>	2740, 2746, 2793
<code>\@pdfborderstyle</code>	164	<code>\Fld@name</code>	2699, 2734, 2872, 2995, 3163, 3237
<code>\@savsf</code>	648, 670	<code>\Fld@onblur@code</code>	3205, 3207
<code>\@schapter</code>	3615, 3616	<code>\Fld@onclick@code</code>	3330
<code>\@secondoftwo</code>	3671	<code>\Fld@onenter@code</code>	3217, 3219
<code>\@sect</code>	3666, 3667	<code>\Fld@onexit@code</code>	3221, 3223
<code>\@spart</code>	3658, 3659	<code>\Fld@onfocus@code</code>	3201, 3203
<code>\@ssect</code>	3601, 3602	<code>\Fld@onmousedown@code</code>	3209, 3211
<code>\@tempdima</code>	2741, 2748, 2750, 2787, 2788, 2789, 2793, 2794, 2795, 2796	<code>\Fld@onmouseup@code</code>	3213, 3215
<code>\@tempdimb</code>	2747, 2748	<code>\Fld@pageobjref</code>	2668, 2683, 3120, 3252, 3305, 3338, 3374, 3413, 3462, 3497
<code>\@typeset@protect</code>	3585	<code>\Fld@radiosymbol</code>	3393
<code>\BKM@color</code>	432, 438	<code>\Fld@rotation</code>	3267, 3269, 3313, 3320, 3322, 3344, 3350, 3352, 3381, 3383, 3421, 3430, 3432, 3468, 3473, 3475, 3503, 3508, 3510
<code>\c@secnumdepth</code>	3635, 3653, 3668	<code>\Fld@submitflags</code>	3489
<code>\calc@bm@number</code>	3577	<code>\Fld@validate@code</code>	3193, 3195
<code>\check@bm@number</code>	3576	<code>\Fld@value</code>	2701, 2717, 2736, 2777, 2778, 3450
<code>\define@key</code>	429	<code>\Fld@width</code>	2702, 2716, 2723, 2727, 2737, 2776, 2787, 2789, 2834, 2848, 2866, 2884, 2914, 2925, 2962, 2974, 2998, 3008, 3043
<code>\Fld@additionalactions</code>	3227, 3297, 3331, 3367, 3404, 3451, 3491, 3520	<code>\Fld@X@additionalactions</code>	3184, 3229, 3232
<code>\Fld@align</code>	3256, 3342, 3417	<code>\Form@action</code>	3487
<code>\Fld@altname</code>	3164, 3166, 3238, 3240	<code>\H@old@part</code>	3645, 3656
<code>\Fld@annotflags</code>	3251, 3304, 3337, 3373, 3412, 3461, 3496	<code>\H@old@schapter</code>	3615, 3624
<code>\Fld@annotnames</code>	3236, 3253, 3306, 3339, 3375, 3414, 3463, 3498	<code>\H@old@sect</code>	3666, 3677, 3686
<code>\Fld@bcolor</code>	3275, 3277, 3358, 3360, 3389, 3391, 3425, 3438, 3440	<code>\H@old@spart</code>	3658, 3664
<code>\fld@bcolor</code>	3346	<code>\H@old@ssect</code>	3601, 3606
<code>\Fld@bordercolor</code>	3271, 3273, 3315, 3324, 3326, 3345, 3354, 3356, 3385, 3387, 3423, 3434, 3436, 3469, 3477, 3479, 3504, 3512, 3514	<code>\href@</code>	275, 281
<code>\Fld@borderstyle</code>	3257, 3310, 3343, 3379, 3418, 3467, 3518	<code>\href@split</code>	281, 282
<code>\Fld@borderwidth</code>	2794, 2795, 3257, 3310, 3343, 3379, 3418, 3467, 3518	<code>\Hy@abspage</code>	622
<code>\Fld@calculate@code</code>	2579, 3197, 3199	<code>\Hy@activeanchorfalse</code>	693
<code>\Fld@calculate@sortkey</code>	2590	<code>\Hy@activeanchortrue</code>	684
<code>\Fld@cbsymbol</code>	3279	<code>\Hy@AtBeginDocument</code>	2544, 2560
<code>\Fld@charsize</code>	2793, 3283, 3364, 3395, 3446, 3502	<code>\Hy@bookmarkstype</code>	1597
<code>\Fld@checkequals</code>	2745, 2812	<code>\Hy@chapapp</code>	3619, 3630
<code>\Fld@choices</code>	3366	<code>\Hy@colorlink</code>	35
<code>\Fld@color</code>	3284, 3286, 3365, 3396, 3447	<code>\Hy@currentbookmarklevel</code>	3558
<code>\Fld@default</code>	2700, 2717, 2735, 2778, 2807, 2808, 2996, 3397, 3401, 3402, 3449		

\Hy@DisableOption	173	\HyAnn@Count	2671, 2672, 2675, 2676, 2677, 2678, 2679, 2680, 2685, 2689
\Hy@drafttrue	1528	\HyField@AddToFields	2573, 2728, 2850, 2867, 2898, 2954, 2983, 3047
\Hy@escapeform	2721, 2818, 2862, 2890, 2929, 2976, 3014, 3122, 3129, 3132	\HyField@afields	2542
\Hy@escapestring	3124, 3126, 3131, 3134, 3135, 3138, 3139, 3148, 3151, 3187, 3191, 3195, 3199, 3203, 3207, 3211, 3215, 3219, 3223, 3279, 3330, 3393, 3449, 3450, 3487	\HyField@AfterAuxOpen	2544, 2570, 2603
\Hy@finaltrue	1533	\HyField@AuxAddToCoFields	2555, 2567, 2588
\Hy@FormObjects	2620, 3055, 3117	\HyField@AuxAddToFields	2548, 2564, 2606
\Hy@gtemp	3180, 3182	\HyField@cofields	2543
\Hy@href	275	\HyField@FlagsCheckBox	3005
\Hy@href@nextactionraw	239	\HyField@FlagsChoice	2786
\Hy@href@page	227	\HyField@FlagsPushButton	2881, 2921, 2972
\Hy@linkfileprefix	1542	\HyField@FlagsRadioButton	2782
\Hy@linktoc	1553	\HyField@FlagsSubmit	2922
\Hy@MakeCurrentHref	200	\HyField@FlagsText	2715
\Hy@MakeCurrentHrefAuto	3603, 3619, 3630, 3660, 3674	\HyField@PDFChoices	2859
\Hy@next	3629, 3638, 3641	\HyField@SetKeys	2711, 2752, 2755, 2774, 2874, 2917, 2965, 3000
\Hy@numberline	166, 3559	\HyPat@ObjRef	3531, 3538, 3545, 3552
\Hy@org@chapter	3627, 3642	\hyper@@link	283
\Hy@OutlineName	3575	\hyper@anchor	672
\Hy@OutlineRerunCheck	3573	\hyper@anchorend	672, 3594, 3621, 3632, 3662
\Hy@pdfmajorversion	168	\hyper@anchorstart	672, 3594, 3621, 3632, 3662
\Hy@pdfminorversion	167	\hyper@link	35, 722
\Hy@pdfstringtrue	30, 593	\hyper@linkend	35, 766
\Hy@pdfversion	3175	\hyper@linkfile	325, 833
\Hy@pstringdef	603, 3155	\hyper@linklaunch	39, 342, 899, 958
\Hy@PutCatalog	610	\hyper@linknamed	40, 176, 964
\Hy@raisedlink	3593, 3620, 3631, 3661	\hyper@linkstart	35, 746
\Hy@RestoreLastskip	669	\hyper@linkurl	308, 369, 786
\Hy@ReturnAfterFi	7, 3156	\hyper@normalise	275, 304, 320, 337, 363, 376
\Hy@safe@activestru	806, 2582	\HyPL@Labels	612, 622
\Hy@SaveLastskip	649	\HyPL@storePageLabel	610
\Hy@secnum@part	3648, 3650, 3653	\HyPsd@SanitizeForOutFile	3562
\Hy@SectionAnchorHref	3584, 3608, 3679	\if@filesw	2562, 2585, 2604
\Hy@SectionHShift	3583, 3592, 3604, 3675	\ifFld@combo	2761, 2791
\Hy@StepCount	2746, 2814	\ifFld@hidden	2716, 2776
\Hy@temp	2583, 2598, 3137, 3138	\ifFld@multiline	2704
\Hy@unicodefalse	3177	\ifFld@popdown	2762
\Hy@VerboseAnchor	650	\ifFld@radio	2758, 2781
\Hy@VerboseLinkStart	729, 753	\ifHy@implicit	3579
\Hy@VerboseLinkStop	737, 774, 816, 884, 944, 988	\ifHy@pdfescapeform	3123, 3133
\Hy@VersionChecked	159	\ifHy@unicode	3176
\Hy@WrapperDef	646	\kv@set@family@handler	179
\Hy@xspace@end	734, 815, 883, 943, 987	\m@ne	3635, 3650
\HyAnn@AbsPageLabel	2667, 2673, 2720, 2817, 2861, 2889, 2928, 2975, 3013	\OBJ@OCG@view	165
		\p@	2750
		\pdf@ifdraftmode	2618

<code>\PDF@SetupDoc</code>	160
<code>\PDFForm@Check</code>	3014, 3046, 3248
<code>\PDFForm@List</code>	2862, 2865, 3335
<code>\PDFForm@Name</code>	.....
	2714, 2775, 2878, 3004, 3162
<code>\PDFForm@Push</code>	2890, 2896, 3301
<code>\PDFForm@Radio</code>	2818, 2838, 3370
<code>\PDFForm@Reset</code>	2976, 2982, 3494
<code>\PDFForm@Submit</code>	2929, 2948, 3459
<code>\PDFForm@Text</code>	2721, 2726, 3409
<code>\protected@edef</code>	698
<code>\strip@pt</code>	3283, 3364, 3395, 3446, 3502
<code>\toks@</code>	3606, 3612, 3677, 3683
<code>\tw@</code>	3607, 3612, 3678, 3683
<code>\url@</code>	376, 379
<code>\Url@def</code>	377
<code>\Url@HyperHook</code>	377
<code>\XR@ext</code>	1539
<code>\z@</code>	376, 2672, 2741, 3344, 3350,
	3381, 3468, 3473, 3503, 3508, 3648
<code>\zref@labelbylist</code>	2677
<code>\zref@labelbyprops</code>	2676
<code>\zref@refused</code>	2678
<code>\texorpdfstring</code>	30
text commands:	
<code>\text_expand:n</code>	66, 841, 2230, 2273
<code>\text_purify:n</code>	8, 580
<code>\textsf</code>	2936, 2940
<code>\the</code>	622,
	2676, 2677, 2678, 2679, 2680, 2685,
	2689, 2789, 2793, 2796, 3612, 3683
<code>\theH...</code>	12
<code>\ThisShouldNotHappen</code>	3142
<code>\title</code>	2
tl commands:	
<code>\c_space_tl</code>	166, 1252,
	1283, 1288, 1293, 2843, 2844, 2851,
	3066, 3068, 3078, 3080, 3082, 3085,
	3087, 3283, 3286, 3418, 3446, 3447
<code>\tl_const:Nn</code>	476
<code>\tl_gput_right:Nn</code>	622
<code>\tl_gset:Nn</code>	1991, 2011, 2015
<code>\tl_if_blank:nTF</code>	.....
	431, 860, 2275, 2312, 2327, 2435, 2448
<code>\tl_if_empty:nTF</code>	.....
	699, 1580, 2232, 3271, 3275, 3284, 3447
<code>\tl_if_empty:nTF</code>	.....
	233, 1117, 1157, 1176,
	1203, 1222, 1766, 1822, 1929, 2214
<code>\tl_if_empty_p:N</code>	1993, 2018
<code>\tl_if_eq:NnTF</code>	2140
<code>\tl_if_exist:NTF</code>	3324, 3434, 3438
<code>\tl_if_exist_p:N</code>	3315, 3423, 3425
<code>\tl_if_head_eq_charcode:nTF</code>	.....
	1006, 1023
<code>\tl_new:N</code>	.....
	206, 207, 208, 209, 465, 466, 470,
	471, 472, 473, 496, 497, 498, 499,
	500, 501, 502, 503, 504, 512, 546, 672
<code>\tl_set:Nn</code>	226,
	227, 239, 270, 273, 299, 302, 358,
	361, 388, 506, 507, 508, 509, 510,
	511, 678, 687, 731, 755, 841, 1976,
	1980, 1984, 2007, 2142, 2145, 2152,
	2155, 2156, 2157, 2158, 2159, 2160,
	2165, 2169, 2205, 2230, 2252, 2273
<code>\tl_set_eq:NN</code>	705
<code>\tl_to_str:N</code>	843, 849, 854
<code>\tl_to_str:n</code>	21, 22, 32, 33, 34, 46
token commands:	
<code>\token_to_str:N</code>	42
<code>\toks</code>	3607, 3612, 3678, 3683

  

<b>U</b>	
<code>unicode</code> (hypersetup key)	1509
<code>\Url</code>	4, 5, 210
<code>url</code> (hypersetup key)	10, 1568
<code>\url</code>	4-6, 22, 350, 376
<code>urlborder</code> (hypersetup key)	14
<code>urlborderstyle</code> (hypersetup key)	14, 1197
<code>urlcolor</code> (hypersetup key)	1083
<code>\urldef</code>	5, 22, 376, 378
<code>\Urlfont</code>	6
<code>\UrlSpecial</code>	4
use commands:	
<code>\use:n</code>	953
<code>\use_i:nn</code>	635, 641
<b>V</b>	
<code>verbose</code> (hypersetup key)	1519
<b>W</b>	
<code>\write</code>	2563, 2566, 2586, 2605
<b>X</b>	
<code>\x</code>	2753, 2773, 3611, 3613, 3682, 3684