

# The hyperref-generic module

## A generic driver for hyperref

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

Version 0.95c, released 2021-03-17

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

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

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

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

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

## 1 Avoiding transition problems

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

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

---

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

- This driver uses `l3color` 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`. This driver uses `l3color` for the colors, some special color models of `xcolor` won't work.
- Load a color package or `graphicx` to get the right page sizes.
- Report problems! Only known problem can be resolved.

## 2 Bookmarks / outlines

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

## 3 “Metadata”

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

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

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

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

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

## 4 Dates

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

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

## 5 PDF page size (mediabox)

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

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

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

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

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

## 6 Commands to create “external” references

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

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

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

For the printing `\url` and `\nolinkurl` rely on the `url` package and its `\Url` command.

(Expandable) commands are expanded and special chars can also be input by commands but beside this no conversion is done: for all input `hyperref` basically assumes that

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

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

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

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

## 6.1 Special problem: links to files

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

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

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

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

## 6.2 Splits

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

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

For this reason three new commands have been create:

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

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

## 6.3 Options

All `\href` commands and `\url` have an option argument for keyval syntax. It accepts the following keys. Not all keys make sense for all keys, but they don't error, they are silently ignored.

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

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

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

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

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

Some main changes are

- The default colors have been changed.
- `citecolor` is no longer set with `allcolors` and it is not part of the color scheme, it is only supported for compability.
- a number of color schemes have been predefined.

### 7.1 Background information

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

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

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

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

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

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

## 7.2 New Keys

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

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc
- Beside `pdfhighlight` there are also `linkhighlight`, `urlhighlight` etc
- Beside `pdfborderstyle` there are also `linkborderstyle`, `urlborderstyle` etc
- Beside `colorlinks` there are also `colorlink`, `colorurl` etc
- Beside `ocgcolorlinks` there are also `ocgcolorlink`, `ocgcolorurl`, etc TODO
- Beside `hidelinks` there are also `hidelink`, `hideurl`, etc
- `bordercolormodel` allows to set the model used in annotations, the allowed values are `rgb` or `cmk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmk` (4 numbers) in the `/C` key of an annotation, this is often ignored by pdf viewers and the colors can be wrong.

- The boolean keys `url`, `link`, `run`, `menu`, `file` allow to deactivate locally the link types.

**colorscheme** (*setup key*) The new key `colorscheme` allows to switch the colors (both for text and borders) with a key word. It takes one of the values `original` (the colors as `hyperref` uses normally), `phelype`, `daleif`, `szabolcsA`, `szabolcsB`, `tivv`, `julian`, `henryford`.

The names refer to the authors in answers and comments in <https://tex.stackexchange.com/questions/525261/better-default-colors-for-hyperref-links>.

The default is `phelype`.

### 7.3 Public interfaces

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

```
\l_hyp_annot_colorlink_bool,
\l_hyp_annot_colorurl_bool,
\l_hyp_annot_colorfile_bool,
\l_hyp_annot_colorruncolor_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorruncolor_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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

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

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

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

### 7.4 Changed behaviour

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

Color keys accept the following input syntax:

```
model based      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.
- `destlabels` (destination names are taken from `\label` if possible)
- `encap`



- **hyperfigures** (according to the **hyperref** manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- **hyperfootnotes**
- **hyperindex**
- **implicit** (redefine L<sup>A</sup>T<sub>E</sub>X 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.

**url** (*setup key*)

**file** (*setup key*)

**run** (*setup key*)

**menu** (*setup key*)

## 11 Draftmode

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

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

## 12 Dropped options

A number of options are ignored by this driver

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

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

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

**unicode** This is always true.

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

```
\DeclareDocumentMetadata{standard=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 pdf-tex and luatex the automatic calculation of the coordinates from the encompassing box. With dvips and (x)dvipdfmx it will fall back to `Fit`. `pdfstartview` and `pdfremotestartview` will fallback to `Fit` too.

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

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

## 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 `cmymk` (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 is 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)

`linkborder` (setup key)

`urlborder` (setup key)

`runborder` (setup key)

`menuborder` (setup key)

This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should according to the reference set round corners, but PDF viewer seem to ignore it. The third number is the line width of the border. Settings done with `pdfborderstyle` should take precedence.

`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]
```

`urlborderstyle` (setup key)

Key	Values	comment / example
<code>/Type</code>	<code>/Border</code>	optional
<code>/W</code>	<code>&lt;number&gt;</code>	Width of border line
<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])

`fileborderstyle` (setup key)

`runborderstyle` (setup key)

`menuborderstyle` (setup key)

`pdfcreationdate` (setup key)

Setting these keys is normally not needed. If they are used the values are stored directly in the Info dictionary for `/Creationdate` and `/ModDate`. 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'`.

`pdfmoddate` (setup key)

`pdflinkmargin` (setup key)

As described in the `hyperref` manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf<sub>l</sub>atex and lua<sub>l</sub>atex work by page, with dvipdfmx the setting is global (and has to be done in the preamble).

`pdflang` (setup key)

The key will work, but it is recommended to the set the language in `\DeclareDocumentMetadata` instead.

## File I

# hyperref-generic driver implementation

```
1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2021-03-17 v0.95c %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
7 \ExplSyntaxOn
```

```

8 \file_input:n {hyperref-colorschemes.def}
9 \ExplSyntaxOff

```

## 1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

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

```

The pdfversion should be set in \DeclareDocumentMetadata

```

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

```

A generic message for ignored keys.

```

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

```

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

```

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

```

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

```
56 \msg_new:nnn
57 { hyp }
58 { empty-destination-name }
59 {
60   Empty~destination~name,\\
61   using~'#1'
62 }
```

A message if the destination check fails

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

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

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

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

```
91 \msg_new:nnn
92 { hyp }
93 { unknown-key }
94 {
95   unknown~key~#2~of~module~'#1'~set~to~'#3'.
96 }
```

```

97 \msg_new:nnn
98 { hyp }
99 { unknown-key-to-Hyp }
100 {
101   ignored-in-family-Hyp-unknown-key-#1.
102 }

```

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

```

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

```

## 2 Variants

```

134 \cs_generate_variant:Nn\pdf_destination:nn {nf}
135 \cs_generate_variant:Nn\pdf_object_ref:n {e}

```

## 3 Overwriting/providing commands from hyperref

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

```

136 \chardef\Hy@VersionChecked=1 %don't check the version!
137 %\cs_set_protected:Npn \PDF@SetupDoc{}
138 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```



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

```
139 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n
```

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

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

```
140 \providecommand\@pdfborder{}
141 \providecommand\@pdfborderstyle{}
142 \newcommand\OBJ@OCG@view {} % needed in hyperref
143 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark
```

We force unicode as option. As bookmarks are still using `\pdfstringdef` we also need to load `puenc.def`.

```
144 \HyPsd@LoadUnicode
145 \Hy@unicodetrue
146 \let\HyPsd@pdfencoding\HyPsd@pdfencoding@unicode
147 \Hy@DisableOption{unicode}
```

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

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

`\Acrobatmenu` should use the new internal link command

```
155 \RenewDocumentCommand \Acrobatmenu { m m }
156 {
157   \hyper@linknamed {#1} {#2}
158 }
```

`\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!

```
159 % TODO should go at some time ...
160 % \kv@set@family@handler{Hyp}
161 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
162 \cs_set_protected:Npn \hypersetup #1
163 {
164   %\kvsetkeys{Hyp} {#1}
165   \keys_set:nn { hyp / setup }{ #1 }
166 }
167 % TODO for now unknown keys should only give warnings.
168 \keys_define:nn { hyp / setup }
169 {
170   unknown .code:n =
171   {
172     \msg_warning:nnxxx { hyp } { unknown-key }
173     { hyp~/~setup }{ \l_keys_key_str } { #1 }
174   }
175 }
```

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`)

```

176
177 \bool_new:N \l__hyp_href_url_encode_bool
178 \bool_new:N \l__hyp_href_url_ismap_bool
179 \tl_new:N \l__hyp_href_url_protocol_tl
180 \tl_new:N \l__hyp_href_pdf_destination_tl
181 \tl_new:N \l__hyp_href_pdf_page_tl
182 \tl_new:N \l__hyp_href_run_parameter_tl
183 \cs_new_protected:Npn \__hyp_href_url_format {\begingroup\Url}
184
185
186 \keys_define:nn { hyp / href }
187 {
188   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
189   ,format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 },
190   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
191   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
192   ,pdfremotestartview .code:n =
193     {
194       \keys_set:nn { hyp / setup }
195       { pdfremotestartview = #1 }
196     }
197   ,page .code:n =
198     {
199       \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
200       \tl_set:Nn \Hy@href@page {#1}
201     }
202   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
203   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
204   ,nextactionraw .code:n =
205     { %perhaps some safety match later, see hyperref code
206       \tl_if_empty:nTF
207       {
208         \pdfdict_remove:nn{l_hyp/annot/A}{Next}
209       }
210       {
211         \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
212         \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
213         \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
214       }
215     }
216   ,afrelationship .code:n =
217     {
218       \pdfdict_put:nnx
219       { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
220     }
221 }
222
223
224 \keys_define:nn { hyp / setup }

```

```

225 {
226   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
227   ,href / urlencode .default:n = {true}
228   ,href / urlencode .initial:n = {false}
229   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
230   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
231   ,href / format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
232 }
233
234
235
236 \DeclareRobustCommand*{\href}[1][\%
237   \begingroup
238   \keys_set:nn { hyp / href } {#1}
239   \let\protect\@unexpandable@protect
240   \bool_if:NTF \l__hyp_href_url_encode_bool
241     {
242       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
243     }
244     {
245       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
246     }
247   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
248 }
249
250
251 \DeclareRobustCommand*{\hrefurl}[1][\{
252   \group_begin:
253   \keys_set:nn { hyp / href } {#1}
254   \let\protect\@unexpandable@protect
255   \bool_if:NTF \l__hyp_href_url_encode_bool
256     {
257       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
258     }
259     {
260       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
261     }
262   \hyper@normalise\__hyp_href_url_aux:nn}
263
264 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
265 {
266   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
267   \group_end:
268 }
269
270
271 \DeclareRobustCommand*{\hrefpdf}[1][\{
272 {
273   \begingroup
274   \keys_set:nn { hyp / href } {#1}
275   \let\protect\@unexpandable@protect
276   \hyper@normalise\__hyp_href_pdf_aux:nn
277 }
278

```

```

279 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
280 {
281   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}\endgroup
282 }
283
284 \DeclareRobustCommand*\hrefrun}[1] []
285 {
286   \group_begin:
287   \keys_set:nn { hyp / href } {#1}
288   \let\protect\@unexpandable@protect
289   \hyper@normalise\__hyp_href_run_aux:nn
290 }
291
292 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
293 {
294   \exp_args:Nnno\hyper@linklaunch{#2}{#1}{\l__hyp_href_run_parameter_tl}
295   \group_end:
296 }
297
298 \DeclareRobustCommand*\url}[1] []
299 {
300   \group_begin:
301   \keys_set:nn { hyp / href } {#1}
302   \let\protect\@unexpandable@protect
303   \bool_if:NTF \l__hyp_href_url_encode_bool
304   {
305     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
306   }
307   {
308     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
309   }
310   \hyper@normalise\__hyp_href_url_aux:n
311 }
312
313 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
314 {
315   \exp_args:Nno
316   \hyper@linkurl{\__hyp_href_url_format {#1}}
317   {\l__hyp_href_url_protocol_tl#1}
318   \group_end:
319 }
320
321
322 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
323 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
324 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
325 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

## 4 Compability 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.

```
326 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
327 {
328   %\tl_set:cn {@#1}{#2}
329   \AddToDocumentProperties[hyperref]{#1}{#2}
330 }
331 \cs_generate_variant:Nn \__hyp_store_metadata:nn {xn,nx,xx}
```

### 4.2 citecolor

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

```
332 \hook_new:n{hyp/link/cite}
333 \color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
334 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
335 \keys_define:nn { hyp / setup }
336 {
337   ,citecolor .code:n = {\__hyp_color_set:nn {hyp/color/cite}{#1}}
338   ,citebordercolor
339     .code:n = {\__hyp_color_set:nn {hyp/color/citeborder}{#1}}
340 }
341 \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
342 {
343   \keys_set:nn { hyp / setup }
344   {
345     ,linkbordercolor= hyp/color/citeborder
346     ,linkcolor      = hyp/color/cite
347   }
348 }
```

## 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

```
349 \bool_lazy_and:nnF
350 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
351 { \msg_error:nn { hyp}{ missing-resource-management } }
```

Outlines/bookmarks require the bookmark package. TODO check pdfpagemode if bookmarks are suppressed.

```
352 \legacy_if:nT { Hy@bookmarks }
353 {
354   \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
355 }
356 \legacy_if:nT { Hy@draft }
```

```

357 {
358   \PassOptionsToPackage{draft}{bookmark}
359 }

```

## 6 Reference and label commands

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

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

```

\__hyp_ref_label:nn
\__hyp_ref_if_exist:nn
\__hyp_ref_check:nn
\__hyp_ref_value:nn
360 %
361 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
362 {
363   \@bsphack
364   \ref_label:nn{#1}{#2}
365   \@esphack
366 }
367 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

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

```

368 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
369 \prg_generate_conditional_variant:Nnn \__hyp_ref_if_exist:nn {en} { p , T , F , TF }

```

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

```

370 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
371 {
372   \__hyp_ref_if_exist:nnF {#1}{#2}
373   {
374     \protect\G@refundefinedtrue
375     \@latex@warning
376     {
377       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
378       on~page~\thepage~\space undefined
379     }
380   }
381 }
382 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

383 \cs_new:Npn \__hyp_ref_value:nn #1 #2
384 {
385   \ref_value:nn{#1}{#2}
386 }
387 \cs_generate_variant:Nn \__hyp_ref_value:nn {en}

```

*(End definition for `\__hyp_ref_label:nn` and others.)*

## 7 Variables

### 7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq 388 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 389 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 390 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 391 \int_new:N \l__hyp_tmpa_int
392 \str_new:N \l__hyp_tmpa_str

```

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

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

```

\l__hyp_dest_name_tmpa_tl 393 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl 394 \tl_new:N \l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl 395 \tl_new:N \l__hyp_filename_tmpa_tl
\__hyp_text_tmpa_str\__g__hyp_text_tmpa_str 396 \tl_new:N \l__hyp_para_tmpa_tl
397 \str_new:N \l__hyp_text_tmpa_str
398 \str_new:N \g__hyp_text_tmpa_str

```

TODO: document and check use!

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

### 7.2 Constants

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

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

(End definition for `\c__hyp_dest_undefined_tl`.)

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

```

400 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
401 {url,link,file,menu,run}
402 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
403 {
404     URI      = url,
405     GoTo     = link,
406     GoToR    = file,
407     Named    = menu,
408     Launch= run
409 }
410 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
411 {
412     url      = URI,
413     link     = GoTo,
414     file     = GoToR,
415     menu     = Named,
416     run      = Launch
417 }
418

```

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

### 7.3 Variables

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

```
419 \tl_new:N \g__hyp_dest_pdfstartpage_tl
420 \tl_new:N \g__hyp_dest_pdfstartview_tl
421 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

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

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

```
\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_tl
\l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_file_print_tl
\l__hyp_text_enc_para_print_tl
422 \tl_new:N \l__hyp_text_enc_uri_print_tl
423 \tl_new:N \l__hyp_text_enc_info_print_tl
424 \tl_new:N \l__hyp_text_enc_dest_tl
425 \tl_new:N \l__hyp_text_enc_dest_print_tl
426 \tl_new:N \l__hyp_text_enc_file_print_tl
427 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```
428
429 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
430 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
431 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
432 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
433 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
434 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}
```

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

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

```
435 \tl_new:N \l__hyp_dest_pdfview_tl
```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

`hyp/annot/url` (color name)

`hyp/annot/file` (color name)

`\g__hyp_bordercolormodel_str` This holds the export model for border color etc. It is currently either `space-sep-cmyk`

`hyp/annot/run` (color name) or `space-sep-rgb`. The default is the second. It can be change by the key

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

```
436 \str_new:N \g__hyp_bordercolormodel_str
```

(End definition for `\g__hyp_bordercolormodel_str`.)



## 7.4 Booleans

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

```
\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
437 \seq_map_inline:Nn \c__hyp_annot_types_seq
438 {
439   \bool_new:c {l_hyp_annot_color#1_bool}
440 }
```

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

These booleans are needed to control the ocgcolors. They are public so that other packages can query the state too.

```
\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
441 \seq_map_inline:Nn \c__hyp_annot_types_seq
442 {
443   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
444 }
```

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

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

```
not_Named_bool_1111\l_hyp_annot_Launch_bool
445 \seq_map_inline:Nn \c_pdfannot_link_types_seq
446 {
447   \bool_new:c {l__hyp_annot_#1_bool}
448   \bool_set_true:c {l__hyp_annot_#1_bool}
449 }
```

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

## 7.5 Boxes

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

```
450 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

## 7.6 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.

```
\c__hyp_dest_startview_regex
451 \regex_const:Nn \c__hyp_dest_startview_regex
452 {
453   \A\ *
454   (?
455     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
456     |
457     (?:Fit\b|FitB\b)
458     |
459     (?:\d+|\d*\.\d+)|\ +null){1}
460     |
461     (?:FitR (?:\ +\d+|\ +\d*\.\d+)){4}\ )
462   )
463 }
```

(End definition for `\c__hyp_dest_startview_regex`.)

## 7.7 PDF dictionaries

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

```
464 \pdfdict_new:n {l__hyp_page/Trans}
465 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End definition for `l__hyp_page/Trans`.)

## 8 PDF string conversion

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

`hyp/text/pdfstring`

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

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

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

```
\__hyp_text_purify:nN
467 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
468 {
469   \str_set:Nx #2 {\text_purify:n { #1 } }
470 }
```

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

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

```
\__hyp_text_cleanup:N
471 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
472 {
473
474 }
```

(End definition for `\__hyp_text_cleanup:N`.)

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

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

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

```

475 \cs_new_protected:Npn \_hyp\_text\_string\_from\_unicode:nN #1 #2
476 {
477   \pdf\_string\_from\_unicode:nVN { #1 } #2 #2
478 }

```

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

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

`\_hyp\_text\_pdfstring:nnN`

```

479 \cs_new_protected:Npn \_hyp\_text\_pdfstring:nnN #1 #2 #3
480 {
481   \group\_begin:
482   \Hy@pdfstringtrue
483   \hook\_use:n {hyp/text/pdfstring}
484   \_hyp\_text\_purify:nN { #1 } \l\_hyp\_text\_tmpa\_str
485   \_hyp\_text\_cleanup:N \l\_hyp\_text\_tmpa\_str
486   \_hyp\_text\_string\_from\_unicode:nN { #2 } \l\_hyp\_text\_tmpa\_str
487   \str\_gset\_eq:NN \g\_hyp\_text\_tmpa\_str\l\_hyp\_text\_tmpa\_str
488   \group\_end:
489   \str\_set\_eq:NN #3 \g\_hyp\_text\_tmpa\_str
490 }
491 \cs\_generate\_variant:Nn \_hyp\_text\_pdfstring:nnN {xnN,onN,xoN,ooN,noN}

```

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

!!! temporary until all instances are gone

```

492 \cs\_new\_protected:Npn \Hy@pstringdef #1 #2
493 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

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

```

494 \cs\_new\_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
495 {
496   \_hyp\_text\_pdfstring:noN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
497 }

```

(End definition for `\_hyp\_text\_pdfstring\_info:nN`.)

## 9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

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

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

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

**dvipdfm:** identical to xetex

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

**luatex** identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-28 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
498 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
499   {
500     \pdfmanagement_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
501   }
502
503 \def\Hy@PutCatalog #1 {}
504
505
506 \legacy_if:nT { Hy@pdfpagelabels }
507   {
508     \cs_set_protected:Npn \HyPL@StorePageLabel #1
509       {
510         \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
511         \__hyp_PageLabels_gpush:
512       }
513   }
```

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

## 10 Core Hyperref Commands

Every `hyperref` has to define eight core command:

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

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

## 10.1 Anchors / destinations

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

---

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

---

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

```
\__hyp_destination:nn

514 \Hy@WrapperDef \__hyp_destination:nn #1 #2
515 {
516   \mode_if_horizontal:T { \@savs\spacefactor }
517   \Hy@SaveLastskip      %defined in hyperref
518   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
519   \__hyp_text_pdfstring:xoN
520   { \HyperDestNameFilter{#1} }
521   { \l__hyp_text_enc_dest_tl }
522   \l__hyp_tmpa_tl
523   \str_if_eq:nnTF {#2} {fitrbx}
524   {
525     \exp_args:NV
526     \pdf_destination:nnnn \l__hyp_tmpa_tl
527     { \box_wd:N \l__hyp_dest_box }
528     { \box_ht:N \l__hyp_dest_box }
529     { \box_dp:N \l__hyp_dest_box }
530   }
531   {
532     \exp_args:NV
533     \pdf_destination:nf
534     { \l__hyp_tmpa_tl }
535     { #2 }
536   }
537   \Hy@RestoreLastskip %defined in hyperref
538   \mode_if_horizontal:T { \spacefactor\@savs }
539 }
```

(End 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.

```
\hyper@anchor
\hyper@anchorstart
\hyper@anchorend

540 \cs_new_protected:Npn \hyper@anchor #1
541 {
542   \exp_args:NnV
543   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
544 }
```

```

545
546 \cs_new_protected:Npn \hyper@anchorstart #1
547 {
548   \Hy@activeanchortrue
549   \exp_args:NnV
550   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
551 }
552
553 \cs_new_protected:Npn \hyper@anchorend
554 {
555   \Hy@activeanchorfalse
556 }

```

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

## 10.2 GoTo Links

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

At first the internal link commands:

```

557 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
558 {
559   \mode_leave_vertical:
560   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
561   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
562   {
563     \msg_warning:nnx
564     { hyp }
565     { empty-destination-name }
566     { \c__hyp_dest_undefined_tl }
567     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
568   }
569   {
570     \__hyp_text_pdfstring:xoN
571     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
572     { \l__hyp_text_enc_dest_tl }
573     \l__hyp_dest_name_tmpa_tl
574   }
575   \exp_args:No
576   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
577 }
578
579 \cs_new_protected:Npn \__hyp_link_goto_end:
580 {
581   \pdfannot_link_goto_end:
582 }

```

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

---

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

---

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

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

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

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

---

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

---

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

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

`hyperref` adds a group with `\Hy@colorlink`, we move this outside the link so that it groups the context hook too.

```
583
584 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
585 {
586   \bool_if:NTF \l__hyp_annot_GoTo_bool
587   {
588     \Hy@VerboseLinkStart{#1}{#2}
589     \group_begin:
590     \hook_use:n {hyp/link/#1}
591     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
592     \__hyp_link_goto_end:
593     \group_end:
594     \Hy@VerboseLinkStop
595   }
596   {\let\protect\relax#3}}
597 }
598
599 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
600 {
601   \bool_if:NT \l__hyp_annot_GoTo_bool
602   {
603     \Hy@VerboseLinkStart{#1}{#2}% only for debug
604     \group_begin:
605     \hook_use:n {hyp/link/#1}
606     \__hyp_link_goto_begin:nw {#2}
607   }
608 }
609
610 \cs_new_protected:Npn \hyper@linkend
611 {
612   \bool_if:NT \l__hyp_annot_GoTo_bool
613   {
614     \__hyp_link_goto_end:
615     \group_end:
```

```

616         \Hy@VerboseLinkStop
617     }
618 }

```

### 10.3 URI links

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

```

619 \pdfdict_new:n {l_hyp/annot/A/URI}
620 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
621 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
622
623 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
624 {
625     \bool_if:NTF \l__hyp_annot_URI_bool
626     {
627         \group_begin:
628         \__hyp_text_pdfstring:xoN
629             { #2}
630             { \l__hyp_text_enc_uri_print_tl }
631             \l__hyp_uri_tmpa_tl
632         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
633         \bool_if:NT \l__hyp_href_url_ismap_bool
634             {
635                 \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
636             }
637         \cs_set_eq:NN \# \c_hash_str
638         \cs_set_eq:NN \% \c_percent_str
639         \Hy@safe@activetrue
640         \mode_leave_vertical:
641         \pdfannot_dict_put:nnx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
642         \pdfannot_link:nnx { URI }
643         {
644         }
645         {
646             \let\protect\relax
647             #1
648             \Hy@xspace@end
649             \Hy@VerboseLinkStop %where is the start??
650         }
651         \group_end:
652     }
653     {{\let\protect\relax#1}}
654 }
655

```

### 10.4 GoToR Links files

```

656 \pdfdict_new:n {l_hyp/annot/A/GoToR}
657 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
658 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
659
660
661 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname

```



```

662 {
663   \bool_if:NTF \l__hyp_annot_GoToR_bool
664   {
665     \group_begin:
666     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
667     \exp_args:Nx
668     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
669     {
670       \pdfdict_put:nnx { l_pdffile/Filespec } { Subtype } { \pdf_name_from_unicode_e:n {ap
671       \exp_args:Nnox
672       \pdffile_embed_file:nnn
673       {}
674       {\l__hyp_filename_tmpa_tl }
675       {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
676     }
677     \pdfdict_put:nnx
678     {l_hyp/annot/A/GoToR}
679     {F}
680     {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
681     \__hyp_text_pdfstring:nnN
682     { #3 }
683     { \l__hyp_text_enc_dest_print_tl }
684     \l__hyp_dest_name_tmpa_tl
685     \tl_if_blank:eTF {#3}
686     {
687       \pdfdict_put:nnx {l_hyp/annot/A/GoToR} {D}
688       {
689         [
690           \int_eval:n
691           { \int_max:nn {0} { 0\l__hyp_href_pdf_page_tl - 1 }}
692           /\l__hyp_dest_pdfremotestartview_tl
693         ]
694       }
695     }
696     {
697       \pdfdict_put:nno {l_hyp/annot/A/GoToR} {D} { \l__hyp_dest_name_tmpa_tl }
698     }
699     \mode_leave_vertical:
700     \pdf_object_unnamed_write:nx {dict} { \pdfdict_use:n {l_hyp/annot/A/GoToR} }
701     \pdfannot_dict_put:nnx {link/GoToR} {A} { \pdf_object_ref_last: }
702     \pdfannot_link:nxn %expansion??
703     { GoToR }
704     {
705     }
706     {
707       \let\protect\relax
708       #1\Hy@xspace@end
709       \Hy@VerboseLinkStop %where is the start??
710     }
711     \group_end:
712   }
713   {{\let\protect\relax#1}}
714 }

```

## 10.5 Launch links

We define `\hyper@linklaunch` for naming consistency

```

715 \pdfdict_new:n {l_hyp/annot/A/Launch}
716 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
717 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
718
719 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
720 {
721   \bool_if:NTF \l__hyp_annot_Launch_bool
722   {
723     \group_begin:
724     \__hyp_text_pdfstring:nnN
725       { #1 }
726       { \l__hyp_text_enc_file_print_tl }
727       \l__hyp_filename_tmpa_tl
728     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
729     \__hyp_text_pdfstring:noN
730       { #3 }
731       { \l__hyp_text_enc_para_print_tl }
732       \l__hyp_para_tmpa_tl
733     \bool_if:nTF
734     {
735       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
736       ||
737       \pdf_version_compare_p:Nn > {1.9}
738     }
739     {
740       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
741     }
742     {
743       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
744       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
745     }
746     \mode_leave_vertical:
747     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
748     \pdfannot_link:nnx
749       { Launch }
750       {
751         % /A
752         % <<
753         % \pdfdict_use:n {l_hyp/annot/A/Launch}
754         % >>
755       }
756       {
757         \let\protect\relax
758         #2\Hy@xspace@end
759         \Hy@VerboseLinkStop %where is the start??
760       }
761     \group_end:
762   }
763   {{\let\protect\relax#2}}
764 }

```

The actually command used by `hyperref` is `\@hyper@launch` which uses a delimited ar-

gument, because of the color the definition is a bit convoluted.

```

765 \use:x
766 { % filename, anchor text, linkname
767   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
768 }
769 {
770   \hyper@linklaunch {#1}{#2}{#3}
771 }

```

## 10.6 Named links (menu)

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

```

772 \pdfdict_new:n {l_hyp/annot/A/Named}
773 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
774 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
775
776 \cs_new_protected:Npn \hyper@linknamed #1 #2 % #1 action, #2 link text
777 {
778   \bool_if:NTF \l__hyp_annot_Named_bool
779   {
780     \group_begin:
781     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
782     {
783       \mode_leave_vertical:
784       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
785       {\pdf_name_from_unicode_e:n{#1}}
786       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
787       \pdfannot_link:nxn { Named }
788       {
789         /A
790         <<
791         \pdfdict_use:n { l_hyp/annot/A/Named }
792         >>
793       }
794       {
795         #2
796         \Hy@xspace@end
797         \Hy@VerboseLinkStop
798       }
799     }
800     {
801       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
802       #2
803     }
804     \group_end:
805   }
806   {\let\protect\relax#2}}
807 }
808

```

## 11 Link decorations

### 11.1 Functions to export and select colors

We support two input syntax: color expressions and model with values.

---

<code>\_hyp_color_export:nnN</code>	<code>\_hyp_color_export:nnN {&lt;color&gt;} {&lt;export format&gt;} &lt;tlvar&gt;</code>
<code>\_hyp_colormodel_export:nnnN</code>	<code>\_hyp_colormodel_export:nnnN {&lt;color model&gt;}{&lt;value&gt;} {&lt;export format&gt;}</code>

---

This exports a color as space separated numbers as needed in in /C. {<color>} should have either the format [model]{value} or be a color expression. For examples: [rgb]{1,0,.5} or red!50!blue. The export format is either space-sep-cmyk or space-sep-rgb.

`\_hyp_color_export:nnN` Color keys need to parse color expressions. Two input types are supported: color=[rgb]{1,0,.5} and color=red!50!blue. The colors are exported to PDF syntax (space separated numbers).

```

809 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
810 {
811   \tl_if_head_eq_charcode:nNTF {#1}[ %]
812   {
813     \_hyp_colormodel_export:wnnN #1 {#2} #3
814   }
815   {
816     \color_export:nnN {#1} {#2} #3
817   }
818 }
819
820 \cs_new_protected:Npn \_hyp_colormodel_export:wnnN [#1] #2 #3 #4
821 {
822   \color_export:nnnnN {#1}{#2}{#3}#4
823 }
824
825 \cs_generate_variant:Nn \_hyp_color_export:nnN {xVN}

```

(End definition for `\_hyp_color_export:nnN` and `\_hyp_colormodel_export:nnnN`.)

---

<code>\_hyp_color_select:n</code>	<code>\_hyp_color_select:nN {&lt;color&gt;}</code>
-----------------------------------	--

---

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} and color=red!50!blue.

```

826 \cs_new_protected:Npn \_hyp_color_select:n #1
827 {
828   \tl_if_head_eq_charcode:nNTF {#1}[ %]
829   {
830     \_hyp_colormodel_select_aux:wn #1
831   }
832   {
833     \color_select:n {#1}
834   }

```

```

835 }
836
837 \cs_new_protected:Npn \__hyp_colormodel_select_aux:wn [#1] #2
838 {
839   \color_select:nn {#1}{#2}
840 }
841
842 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

---

```

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

```

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

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

```

\__hyp_colormodel_set_aux:nwn
843 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
844 {
845   \tl_if_head_eq_charcode:nNTF {#2}[ %]
846   {
847     \__hyp_colormodel_set_aux:nwn { #1 } #2
848   }
849   {
850     \color_set:nn {#1} {#2}
851   }
852 }
853
854 \cs_new_protected:Npn \__hyp_colormodel_set_aux:nwn #1 [#2] #3
855 {
856   \color_set:nnn {#1}{#2}{#3}
857 }
858
859 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End definition for `\__hyp_color_set:nn` and `\__hyp_colormodel_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.

```

860 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
861 {
862   \hook_gput_code:nnn
863     {pdfannot/link/#2/begin}
864     {hyp/color}
865     {
866       \bool_if:cT { l_hyp_annot_color#1_bool }
867       {
868         \group_begin:
869         \color_select:n { hyp/color/#1}
870       }

```

```

871     }
872     \hook_gput_code:nnn
873     {pdfannot/link/#2/end}
874     {hyp/color}
875     {
876       \bool_if:cT { l_hyp_annot_color#1_bool }
877       {
878         \group_end:
879       }
880     }
881   }

```

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

```

882 \keys_define:nn { hyp / setup }
883 {
884   ,colorlinks .meta:n =
885   {
886     ,pdfborder={0~0~0}
887     ,pdfborderstyle=
888     ,colorurl  =#1
889     ,colorlink =#1
890     ,colorrun  =#1
891     ,colormenu =#1
892     ,colorfile =#1
893   }
894   ,colorlinks .default:n = {true}
895 }

```

**colorurl** (*setup key*)

**colorlink** (*setup key*)

**colorrun** (*setup key*)

**colormenu** (*setup key*)

**colorfile** (*setup key*)

**urlcolor** (*setup key*)

**linkcolor** (*setup key*)

**runcolor** (*setup key*)

**menucolor** (*setup key*)

**filecolor** (*setup key*)

**allcolor** (*setup key*)

```

896 \seq_map_inline:Nn \c__hyp_annot_types_seq
897 {
898   \keys_define:nn { hyp / setup }
899   {
900     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
901     ,#1color .code:n =      { \__hyp_color_set:ne {hyp/color/#1}{##1} }
902   }
903 }
904
905 \keys_define:nn { hyp / setup }
906 {
907   ,allcolors .meta:n =
908   {
909     ,urlcolor=#1
910     ,linkcolor=#1
911     ,runcolor=#1
912     ,filecolor=#1
913     ,menucolor=#1
914   }
915   ,allcolors .value_required:n = true
916 }

```

## 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*)

```
917 \keys_define:nn { hyp / setup }
918 {
919   ,bordercolormodel .choices:nn =
920     {rgb,cmyk}
921     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
922   ,bordercolormodel .initial:n ={rgb}
923 }

924 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
925 {
926   \keys_define:nn { hyp / setup }
927   {
928     #1bordercolor .code:n =
929     {
930       \tl_if_empty:nTF { ##1 }
931       {
932         \pdfannot_dict_remove:nn
933         {link/#2}
934         { C }
935       }
936       {
937         \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
938         \pdfannot_dict_put:nnx
939         {link/#2}
940         { C }
941         { [\l__hyp_tmpa_tl] }
942       }
943     }
944   }
945 }

946
947 \keys_define:nn { hyp / setup }
948 {
949   ,allbordercolors .meta:n =
950   {
951     ,linkbordercolor=#1
952     ,urlbordercolor =#1
953     ,filebordercolor=#1
954     ,menubordercolor=#1
955     ,runbordercolor =#1
956   }
957   ,allbordercolors .value_required:n = true
958 }
959
```

### 11.3.2 Borderwidth and -arc

```

960 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
961 {
962   \keys_define:nn { hyp / setup }
963   {
964     #1border .code:n =
965     {
966       \tl_if_empty:nTF { ##1 }
967       {
968         \pdfannot_dict_remove:nn
969         {link/#2}
970         { Border }
971       }
972       {
973         \pdfannot_dict_put:nnn
974         {link/#2}
975         { Border }
976         { [##1] }
977       }
978     }
979   }
980 }
981 \keys_define:nn { hyp / setup }
982 {
983   ,pdfborder .code:n =
984   {
985     \tl_if_empty:nTF { #1 }
986     {
987       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
988       {
989         \pdfannot_dict_remove:nn
990         {link/##2}
991         { Border }
992       }
993     }
994     {
995       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
996       {
997         \pdfannot_dict_put:nnn
998         {link/##2}
999         { Border }
1000         { [#1] }
1001       }
1002     }
1003   }
1004   ,pdfborder .initial:n = {0~0~1},
1005 }

```

### 11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1006 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1007 {
runborderstyle (setup key) 1008   \keys_define:nn { hyp / setup }
fileborderstyle (setup key)
menuborderstyle (setup key)

```



```

1009     {
1010         #1borderstyle .code:n =
1011         {
1012             \tl_if_empty:nTF { ##1 }
1013             {
1014                 \pdfannot_dict_remove:nn
1015                 {link/#2}
1016                 { BS }
1017             }
1018             {
1019                 \pdfannot_dict_put:nnn
1020                 {link/#2}
1021                 { BS }
1022                 { <<##1>> }
1023             }
1024         }
1025     }
1026 }
1027 \keys_define:nn { hyp / setup }
1028 {
1029     ,pdfborderstyle .code:n =
1030     {
1031         \tl_if_empty:nTF { #1 }
1032         {
1033             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1034             {
1035                 \pdfannot_dict_remove:nn
1036                 {link/##2}
1037                 { BS }
1038             }
1039         }
1040         {
1041             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1042             {
1043                 \pdfannot_dict_put:nnn
1044                 {link/##2}
1045                 { BS }
1046                 { <<##1>> }
1047             }
1048         }
1049     }
1050     ,pdfborderstyle .initial:n = {},
1051 }

```

## 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.

```

1052 \cs_new_protected:Npn \__hyp_ocg_init:
1053 {
1054     \pdf_object_new:nn { l__hyp_ocg_view_dict_obj } { dict }

```

```

1055 \pdf_object_new:nn { l__hyp_ocg_print_dict_obj } { dict }
1056 \pdf_object_new:nn { l__hyp_ocg_config_dict_obj } { dict }
1057 \pdf_object_new:nn { l__hyp_ocg_ref_array_obj } { array }
1058 \pdf_object_write:nx { l__hyp_ocg_ref_array_obj }
1059 {
1060   \pdf_object_ref:n { l__hyp_ocg_view_dict_obj }
1061   \c_space_tl
1062   \pdf_object_ref:n { l__hyp_ocg_print_dict_obj }
1063 }
1064 \pdf_object_write:nn { l__hyp_ocg_view_dict_obj }
1065 {
1066   /Type/OCG
1067   /Name(View)
1068   /Usage
1069   <<
1070     /Print <</PrintState/OFF>>~
1071     /View <</ViewState/ON >>~
1072   >>
1073 }
1074 \pdf_object_write:nn { l__hyp_ocg_print_dict_obj }
1075 {
1076   /Type/OCG
1077   /Name(Print)
1078   /Usage
1079   <<
1080     /Print <</PrintState/ON>>~
1081     /View <</ViewState/OFF>>~
1082   >>
1083 }
1084 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_ref_array_obj} }
1085 \pdfmanagement_add:nxx { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {l__hyp_ocg_print_dict_obj} }
1086 \pdf_object_write:nx { l__hyp_ocg_config_dict_obj }
1087 {
1088   /OFF[\pdf_object_ref:n { l__hyp_ocg_print_dict_obj }]
1089   /AS[
1090     <<
1091       /Event/View
1092       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1093       /Category[/View]
1094     >>
1095     <<
1096       /Event/Print
1097       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1098       /Category[/Print]
1099     >>
1100     <<
1101       /Event/Export
1102       /OCGs\c_space_tl \pdf_object_ref:n { l__hyp_ocg_ref_array_obj }
1103       /Category[/Print]
1104     >>
1105   ]
1106 }
1107 \pdfmanagement_add:nxx { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { l__hyp_ocg_ref_array_obj } }
1108 \cs_gset:Npn \__hyp_ocg_init: {}

```

1109 }

(End definition for `\_hyp_ocg_init:`.)

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

```

1110 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1111 {
1112   \hook_gput_code:nnn
1113     {pdfannot/link/#2/begin}
1114     {hyp/ocg}
1115     {
1116       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1117       {
1118         \_hyp_ocg_init:
1119         \group_begin:
1120         \hbox_set:Nw \l__hyp_tmpa_box
1121       }
1122     }
1123   \hook_gput_code:nnn
1124     {pdfannot/link/#2/end}
1125     {hyp/ocg}
1126     {
1127       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1128       {
1129         \hbox_set_end:
1130         \mbox
1131         {
1132           \pdf_bdcobject:nn {0C}{l__hyp_ocg_print_dict_obj}
1133           \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1134           \pdf_emc:
1135           \pdf_bdcobject:nn {0C}{l__hyp_ocg_view_dict_obj}
1136           \group_begin:
1137           \color_select:n { hyp/color/#1 }
1138           \box_use_drop:N \l__hyp_tmpa_box
1139           \group_end:
1140           \pdf_emc:
1141         }
1142         \group_end:
1143       }
1144     }
1145 }

```

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

```

ocgcolorlink (setup key) 1146 \pdf_version_compare:NnTF < {1.5}
ocgcolorurl (setup key) 1147 {
ocgcolorfile (setup key) 1148   \keys_define:nn { hyp / setup }
ocgcolormenu (setup key) 1149   {
ocgcolorrun (setup key) 1150     ,_ocgcolorlinks .code:n =
1151     {
1152       \msg_warning:nnxx
1153       { hyp }
1154       { ignore-deprecated-or-unknown-option-in-pdf-version }
1155       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1156     }

```

```

1157     }
1158   }
1159   {
1160     \keys_define:nn { hyp / setup }
1161     {
1162       ,_ocgcolorlinks .meta:n =
1163       {
1164         ocgcolorlink=#1,
1165         ocgcolorurl=#1,
1166         ocgcolorfile=#1,
1167         ocgcolorrun=#1,
1168         ocgcolormenu=#1
1169       }
1170       ,_ocgcolorlinks .default:n = true
1171     }
1172   }
1173
1174   \keys_define:nn { hyp / setup }
1175   {
1176     ,ocgcolorlinks .choice:
1177     ,ocgcolorlinks / true .meta:n =
1178     {
1179       pdfborder      = {0~0~0},
1180       pdfborderstyle = {},
1181       colorlinks     = false,
1182       _ocgcolorlinks = true
1183     }
1184     ,ocgcolorlinks / false .meta:n =
1185     {
1186       _ocgcolorlinks = false
1187     }
1188     ,ocgcolorlinks .default:n = {true}
1189   }
1190
1191   \seq_map_inline:Nn \c__hyp_annot_types_seq
1192   {
1193     \pdf_version_compare:NnTF < {1.5}
1194     {
1195       \keys_define:nn { hyp / setup }
1196       {
1197         ,ocgcolor#1 .code:n=
1198         {
1199           \msg_warning:nnxx
1200           { hyp }
1201           { ignore-deprecated-or-unknown-option-in-pdf-version }
1202           { ocgcolor#1 }
1203           { \pdf_version_major:.\pdf_version_minor: }
1204         }
1205       }
1206     }
1207     {
1208       \keys_define:nn { hyp / setup }
1209       {
1210         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }

```

```

1211     }
1212   }
1213 }

```

## 11.5 Highlighting

This keys set what happens if you click on a link

```

1214 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1215 {
1216   \keys_define:nn { hyp / setup }
1217   {
1218     ,#1highlight .choices:nn =
1219     { /I, /N, /O, /P}
1220     {
1221       \pdfannot_dict_put:nnn
1222       {link/#2}
1223       { H }
1224       { ##1 }
1225     }
1226     ,#1highlight / .code:n =
1227     {
1228       \pdfannot_dict_remove:nn
1229       {link/#2}
1230       { H }
1231     }
1232     }
1233     ,#1highlight / unknown .code:n =
1234     {
1235       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1236       { pdfhighlight }
1237       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1238       { \exp_not:n {##1} }
1239     }
1240   }
1241 }
1242 }
1243
1244
1245 \keys_define:nn { hyp / setup }
1246 {
1247   ,pdfhighlight .choices:nn =
1248   { /I, /N, /O, /P}
1249   {
1250     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1251     {
1252       \pdfannot_dict_put:nnn
1253       {link/####2}
1254       { H }
1255       { #1 }
1256     }
1257   }
1258   ,pdfhighlight / .code:n =
1259   {
1260     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop

```

```

1261         {
1262             \pdfannot_dict_remove:nn
1263             {link/##2}
1264             { H }
1265         }
1266     }
1267     ,pdfhighlight .initial:n = {/I},
1268     ,pdfhighlight / unknown .code:n =
1269     {
1270         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1271         { pdfhighlight }
1272         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1273         { \exp_not:n {#1} }
1274     }
1275 }

```

## 11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1276 \keys_define:nn { hyp / setup }
hideurl (setup key) 1277 {
hidefile (setup key) 1278     hidelinks .meta:n =
hiderun (setup key) 1279     {
hidemenu (setup key) 1280         ,colorlinks      = false
1281         ,ocgcolorlinks = false
1282         ,pdfborder      = { 0~0~0 }
1283         ,pdfborderstyle=
1284     }
1285 }
1286
1287 \seq_map_inline:Nn \c__hyp_annot_types_seq
1288 {
1289     \keys_define:nn { hyp / setup }
1290     {
1291         hide#1 .meta:n =
1292         {
1293             ,color#1      = false
1294             ,ocgcolor#1   = false
1295             ,#1border     = { 0~0~0 }
1296             ,#1borderstyle =
1297         }
1298     }
1299 }

```

## 11.7 color schemes and settings

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

```

colorscheme (setup key)
1300 \keys_define:nn { hyp / setup }
1301 {
1302     colorscheme .code:n =

```

```

1303     {
1304         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1305         {
1306             \keys_set:nn { hyp /setup }
1307             {
1308                 ##1 = ##2
1309             }
1310         }
1311     }
1312 }
1313 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

## 12 Keys

### 12.1 Ignored keys

The following are ignored (with or without warnings)

```

unicode (setup key)
pdfencoding (setup key) 1314 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1315 {
1316     ,unicode .code:n = {}
1317     ,pdfencoding .code:n = {}
1318     ,pdfversion .code:n =
1319     {
1320         \msg_warning:nn { hyp }{ pdfversion-disabled }
1321     }
1322 }
1323 %

```

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

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1324 \keys_define:nn { hyp / setup }
draft (setup key) 1325 {
1326     ,verbose .choice:
1327     ,verbose / true .code:n = { \Hy@verbosetrue}
1328     ,verbose / false .code:n = { \Hy@verbosefalse}
1329     ,verbose .default:n = {true}
1330     ,debug .meta:n = {verbose=#1}
1331     ,debug .default:n = {true}
1332 }
1333 \keys_define:nn { hyp / setup }
1334 {
1335     ,draft .code:n =
1336     {
1337         \Hy@drafttrue
1338         \PassOptionsToPackage{draft}{bookmark}
1339     }
1340     ,final .code:n =
1341     {

```

```

1342         \Hy@finaltrue
1343         \PassOptionsToPackage{final}{bookmark}
1344     }
1345 }

```

```

extension (setup key)
hypertextnames (setup key) 1346 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1347 {
    pageanchor (setup key) 1348 ,extension .tl_set:N = \XR@ext
    linktoc (setup key) 1349 ,extension .initial:n= pdf
    linktocpage (setup key) 1350 ,hypertextnames .choice:
    plainpages (setup key) 1351 ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
localanchorname (setup key) 1352 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
linkfileprefix (setup key) 1353 ,hypertextnames .default:n = {true}
1354 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1355 ,localanchorname .choice:
1356 ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1357 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1358 ,localanchorname .default:n = {true}
1359 ,naturalnames .choice:
1360 ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1361 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1362 ,naturalnames .default:n = {true}
1363 ,pageanchor .choice:
1364 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1365 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1366 ,pageanchor .default:n = {true}
1367 ,plainpages .choice:
1368 ,plainpages / true .code:n = { \Hy@plainpagestrue}
1369 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1370 ,plainpages .default:n = {true}
1371 }
1372
1373 \keys_define:nn { hyp / setup }
1374 {
1375 ,linktoc .choices:nn = { none, section, all, page }
1376 {
1377     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1378 }
1379 ,linktoc / unknown .code:n =
1380 {
1381     \msg_warning:nnxxx { hyp } { unknown-choice }
1382     { linktoc }
1383     { none, section, all, page }
1384     { \exp_not:n {#1} }
1385 }
1386 ,linktocpage .choice:
1387 ,linktocpage / true .meta:n = {linktoc=page}
1388 ,linktocpage / false .meta:n = {linktoc=section}
1389 ,linktocpage .default:n = true
1390 }
1391

```

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

`url (setup key)`

`file (setup key)`

`menu (setup key)`

`run (setup key)`



```

1392 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1393 {
1394   \keys_define:nn { hyp / setup }
1395   {
1396     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1397   }
1398 }

1399 \keys_define:nn { hyp / setup }
1400 {
1401   ,baseurl .code:n =
1402   {
1403     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1404     \tl_if_empty:NTF \l__hyp_tmpa_tl
1405     {
1406       \pdfmanagement_remove:nn {Catalog} { URI }
1407     }
1408     {
1409       \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1410     }
1411   }
1412   %only false does something ...
1413   ,bookmarks .choice:
1414   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1415   ,bookmarks / true .code:n = {}
1416   ,bookmarks .default:n = {true}
1417   ,bookmarksnumbered .choice:
1418   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1419   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1420   ,bookmarksnumbered .default:n = {true}
1421   ,bookmarksopen .choice:
1422   ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1423   ,bookmarksopen / true .code:n = { \Hy@bookmarksopentru }
1424   ,bookmarksopen .default:n = {true}
1425   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1426   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1427   ,pdfcenterwindow .choice:
1428   ,pdfcenterwindow / false .code:n =
1429   {
1430     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1431   }
1432   ,pdfcenterwindow / true .code:n =
1433   {
1434     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }{ CenterWindow }{ true }
1435   }
1436   ,pdfcenterwindow / .code:n =
1437   {
1438     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1439   }
1440   ,pdfcenterwindow / unknown .code:n =
1441   {
1442     \msg_warning:nxxx { hyp } { no-bool }
1443     { pdfcenterwindow }
1444     { \exp_not:n {#1} }

```

```

1445     }
1446 ,pdfcenterwindow      .default:n = true
1447 ,pdfdirection        .choice:
1448 ,pdfdirection / L2R .code:n =
1449     {
1450         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1451     }
1452 ,pdfdirection / R2L .code:n =
1453     {
1454         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1455     }
1456 ,pdfdirection /      .code:n =
1457     {
1458         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1459     }
1460 ,pdfdirection / unknown .code:n =
1461     {
1462         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1463         { pdfdirection }
1464         { L2R , R2L }
1465         { \exp_not:n {#1} }
1466     }
1467 ,pdfdisplaydoctitle   .choice:
1468 ,pdfdisplaydoctitle / false .code:n =
1469     {
1470         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1471     }
1472 ,pdfdisplaydoctitle / true .code:n =
1473     {
1474         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1475     }
1476 ,pdfdisplaydoctitle   .default:n = true
1477 ,pdfduplex            .choices:nn =
1478     {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1479     {
1480         \pdf_version_compare:NnTF > {1.6}
1481         {
1482             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1483             { PrintDuplex } { /#1 }
1484         }
1485         {
1486             \msg_warning:nnxx
1487             {hyp}
1488             {ignore-deprecated-or-unknown-option-in-pdf-version}
1489             {pdfduplex}
1490             {\pdf_version:}
1491         }
1492     }%
1493 ,pdfduplex /          .code:n =
1494     {
1495         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1496     }
1497 ,pdfduplex / unknown .code:n =
1498     {

```

```

1499     \msg_warning:nxxx { hyp } { unknown-choice+empty }
1500     { pdfduplex }
1501     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1502     { \exp_not:n {#1} }
1503 }
1504 ,pdffitwindow .choice:
1505 ,pdffitwindow / false .code:n =
1506 {
1507     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1508 }
1509 ,pdffitwindow / true .code:n =
1510 {
1511     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1512 }
1513 ,pdffitwindow / .code:n =
1514 {
1515     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1516 }
1517 ,pdffitwindow .default:n = true
1518 ,pdffitwindow / unknown .code:n =
1519 {
1520     \msg_warning:nxxx { hyp } { no-bool }
1521     { pdffitwindow }
1522     { \exp_not:n {#1} }
1523 }
1524 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1525 ,pdflinkmargin .initial:n = {1pt}
1526 ,pdfmenubar .choice:
1527 ,pdfmenubar / true .code:n =
1528 {
1529     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1530 }
1531 ,pdfmenubar / false .code:n =
1532 {
1533     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1534     { HideMenubar } { true }
1535 }
1536 ,pdfmenubar / .code:n =
1537 {
1538     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1539 }
1540 ,pdfmenubar .default:n = true
1541 ,pdfmenubar / unknown .code:n =
1542 {
1543     \msg_warning:nxxx { hyp } { no-bool }
1544     { pdfmenubar }
1545     { \exp_not:n {#1} }
1546 }
1547 ,pdfnewwindow .choice:
1548 ,pdfnewwindow / true .code:n =
1549 {
1550     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1551     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1552 }

```

```

1553 ,pdfnewwindow / false .code:n =
1554 {
1555     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1556     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1557 }
1558 ,pdfnewwindow / .code:n =
1559 {
1560     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1561     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1562 }
1563 ,pdfnonfullscreenpagemode .choices:nn =
1564 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1565 {
1566     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1567     { NonFullScreenPageMode } {/#1}
1568 }
1569 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1570 {
1571     \pdf_version_compare:NnTF < {1.6}
1572     {
1573         %message
1574     }
1575     {
1576         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1577         {NonFullScreenPageMode}{/UseAttachments}
1578     }
1579 }
1580 ,pdfnonfullscreenpagemode / .code:n =
1581 {
1582     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1583 }
1584 ,pdfnonfullscreenpagemode / unknown .code:n =
1585 {
1586     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1587     { pdfnonfullscreenpagemode }
1588     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1589     { \exp_not:n {#1} }
1590 }
1591 ,pdfnumcopies .code:n =
1592 {
1593     \pdf_version_compare:NnTF > {1.6}
1594     {
1595         \tl_if_empty:NTF {#1}
1596         {
1597             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1598         }
1599         {
1600             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1601             {NumCopies}{#1}
1602         }
1603     }
1604     {
1605         \msg_warning:nnxx
1606         {hyp}

```

```

1607         {ignore-deprecated-or-unknown-option-in-pdf-version}
1608         {pdfnumcopies}
1609         {\pdf_version:}
1610     }
1611 }
1612 ,pdfpagelayout .choices:nn =
1613 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1614 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1615 ,pdfpagelayout / .code:n =
1616 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1617 ,pdfpagelayout / unknown .code:n =
1618 {
1619     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1620     { pdfpagelayout }
1621     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1622     { \exp_not:n {#1} }
1623 }
1624 ,pdfpagemode .choices:nn =
1625 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1626 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1627 ,pdfpagemode / UseAttachments .code:n =
1628 {
1629     \pdf_version_compare:NnTF > {1.5}
1630     {
1631         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1632     }
1633     {
1634         \msg_warning:nnxx
1635         {hyp}
1636         {ignore-deprecated-or-unknown-value-in-pdf-version}
1637         {UseAttachments}
1638         {\pdf_version:}
1639     }
1640 }
1641 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1642 ,pdfpagemode / unknown .code:n =
1643 {
1644     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1645     { pdfpagemode }
1646     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1647     { \exp_not:n {#1} }
1648 }
1649 ,pdfpagescrop .code:n =
1650 {
1651     \tl_if_empty:nTF %or blank?
1652     {
1653         \pdfmanagement_remove:nn {Pages} { CropBox }
1654     }
1655     {
1656         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1657     }
1658 }
1659 ,pdfpicktraybypdfsize .choice:
1660 ,pdfpicktraybypdfsize / true .code:n =

```

```

1661 {
1662   \pdf_version_compare:NnTF > {1.6}
1663   {
1664     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1665     { PickTrayByPDFSize } { true }
1666   }
1667   {
1668     \msg_warning:nnxx
1669     {hyp}
1670     {ignore-deprecated-or-unknown-option-in-pdf-version}
1671     {pdfpicktraybypdfsize}
1672     {\pdf_version:}
1673   }
1674 }
1675 ,pdfpicktraybypdfsize / false .code:n =
1676 {
1677   \pdf_version_compare:NnTF > {1.6}
1678   {
1679     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1680     { PickTrayByPDFSize } { false }
1681   }
1682   {
1683     \msg_warning:nnxx
1684     {hyp}
1685     {ignore-deprecated-or-unknown-option-in-pdf-version}
1686     {pdfpicktraybypdfsize}
1687     {\pdf_version:}
1688   }
1689 }
1690 ,pdfpicktraybypdfsize / .code:n =
1691 {
1692   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1693 }
1694 ,pdfpicktraybypdfsize / unknown .code:n =
1695 {
1696   \msg_warning:nnxx { hyp } { no-bool }
1697   { picktraybypdfsize }
1698   { \exp_not:n {#1} }
1699 }
1700 ,pdfprintarea .choices:nn =
1701 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1702 {
1703   \pdf_version_compare:NnTF < {2.0}
1704   {
1705     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1706     { PrintArea } { /#1 }
1707   }
1708   {
1709     \msg_warning:nnxx
1710     {hyp}
1711     {ignore-deprecated-or-unknown-option-in-pdf-version}
1712     {pdfprintarea}
1713     {\pdf_version:}
1714   }

```

```

1715     }%
1716 ,pdfprintarea / .code:n =
1717 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1718 ,pdfprintarea / unknown .code:n =
1719 {
1720     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1721     { pdfprintarea }
1722     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1723     { \exp_not:n {#1} }
1724 }
1725 ,pdfprintclip .choices:nn =
1726 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1727 {
1728     \pdf_version_compare:NnTF < {2.0}
1729     {
1730         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1731         { PrintClip } { /#1 }
1732     }
1733     {
1734         \msg_warning:nnxx
1735         {hyp}
1736         {ignore-deprecated-or-unknown-option-in-pdf-version}
1737         {pdfprintclip}
1738         {\pdf_version:}
1739     }
1740 }%
1741 ,pdfprintclip / .code:n =
1742 {
1743     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1744 }
1745 ,pdfprintclip / unknown .code:n =
1746 {
1747     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1748     { pdfprintclip }
1749     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1750     { \exp_not:n {#1} }
1751 }
1752 ,pdfprintpagerange .code:n =
1753 {
1754     \pdf_version_compare:NnTF > {1.6}
1755     {
1756         \tl_if_empty:nTF { #1}
1757         {
1758             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1759             { PrintPageRange }
1760         }
1761         {
1762             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1763             {PrintPageRange}{[#1]}
1764         }
1765     }
1766     {
1767         \msg_warning:nnxx
1768         {hyp}

```

```

1769         {ignore-deprecated-or-unknown-option-in-pdf-version}
1770         {pdfprintpagerange}
1771         {\pdf_version:}
1772     }
1773 }
1774 ,pdfprintscaling .choices:nn =
1775 { None, AppDefault }
1776 {
1777     \pdf_version_compare:NnTF > {1.5}
1778     {
1779         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1780         { PrintScaling } { /#1 }
1781     }
1782     {
1783         \msg_warning:nnxx
1784         {hyp}
1785         {ignore-deprecated-or-unknown-option-in-pdf-version}
1786         {pdfprintscaling}
1787         {\pdf_version:}
1788     }
1789 }%
1790 ,pdfprintscaling / .code:n =
1791 {
1792     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1793 }
1794 ,pdfprintscaling / unknown .code:n =
1795 {
1796     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1797     { pdfprintarea }
1798     { None, AppDefault }
1799     { \exp_not:n {#1} }
1800 }
1801 ,pdfremotestartview .code:n =
1802 {
1803     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1804     \exp_args:NNV
1805     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1806     {
1807         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1808     }
1809     {
1810         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1811         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1812     }
1813 }
1814 ,pdfremotestartview .initial:n = {Fit}
1815 % pdfstartpage is special as it shares code with pdfstartview
1816 ,pdfstartpage .code:n =
1817 {
1818     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1819     \bool_if:nTF
1820     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1821     {
1822         \pdfmanagement_remove:nn {Catalog} { OpenAction }

```



```

1823     }
1824     {
1825         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1826         {
1827             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1828         }
1829     }
1830 }
1831 ,pdfstartpage .initial:n =1
1832 ,pdfstartview .code:n =
1833 {
1834     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1835     \exp_args:NNV
1836     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1837     {
1838         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1839     }
1840     {
1841         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1842         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1843     }
1844     \bool_if:nTF
1845     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1846     {
1847         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1848     }
1849     {
1850         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1851         {
1852             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1853         }
1854     }
1855 }
1856 ,pdfstartview .initial:n = Fit
1857 ,pdftoolbar .choice:
1858 ,pdftoolbar / true .code:n =
1859 {
1860     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1861 }
1862 ,pdftoolbar / false .code:n =
1863 {
1864     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1865     { HideToolbar } { true }
1866 }
1867 ,pdftoolbar / true .code:n =
1868 {
1869     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1870 }
1871 ,pdftoolbar .default:n = true
1872 ,pdftoolbar / unknown .code:n =
1873 {
1874     \msg_warning:nnxx { hyp } { no-bool }
1875     { pdftoolbar }
1876     { \exp_not:n {#1} }

```

```

1877     }
1878 % pdfview see below.
1879 ,pdfviewarea .choices:nn =
1880 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1881 {
1882   \pdf_version_compare:NnTF < {2.0}
1883   {
1884     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1885     { ViewArea } { /#1 }
1886   }
1887   {
1888     \msg_warning:nnxx
1889     {hyp}
1890     {ignore-deprecated-or-unknown-option-in-pdf-version}
1891     {pdfviewarea}
1892     {\pdf_version:}
1893   }
1894 }%
1895 ,pdfviewarea / .code:n =
1896 {
1897   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1898 }
1899 ,pdfviewarea / unknown .code:n =
1900 {
1901   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1902   { pdfviewarea }
1903   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1904   { \exp_not:n {#1} }
1905 }
1906 ,pdfviewclip .choices:nn =
1907 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1908 {
1909   \pdf_version_compare:NnTF < {2.0}
1910   {
1911     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1912     { ViewClip } { /#1 }
1913   }
1914   {
1915     \msg_warning:nnxx
1916     {hyp}
1917     {ignore-deprecated-or-unknown-option-in-pdf-version}
1918     {pdfviewclip}
1919     {\pdf_version:}
1920   }
1921 }%
1922 ,pdfviewclip / .code:n =
1923 {
1924   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1925 }
1926 ,pdfviewclip / unknown .code:n =
1927 {
1928   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1929   { pdfviewclip }
1930   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1931     { \exp_not:n {#1} }
1932   }
1933   ,pdfwindowui .choice:
1934   ,pdfwindowui / true .code:n =
1935   {
1936     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1937   }
1938   ,pdfwindowui / false .code:n =
1939   {
1940     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1941     { HideWindowUI } { true }
1942   }
1943   ,pdfwindowui / .code:n =
1944   {
1945     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1946   }
1947   ,pdfwindowui / unknown .code:n =
1948   {
1949     \msg_warning:nnxx { hyp } { no-bool }
1950     { pdfwindowui }
1951     { \exp_not:n {#1} }
1952   }
1953   ,pdfwindowui .default:n = true
1954 }

```

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

```

1955 \keys_define:nn { hyp / setup }
1956 {
1957   ,pdfview .code:n =
1958   {
1959     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
1960     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
1961       { xyz }
1962       {
1963         \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
1964         {
1965           \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
1966           \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
1967           {
1968             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1969           }
1970           {
1971             \tl_set:Nx \l__hyp_dest_pdfview_tl
1972             {
1973               \fp_eval:n { \l__hyp_tmpa_tl * 100 }
1974             }
1975           }
1976         }
1977       }
1978       {
1979         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
1980       }
1981     }
1982     { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }

```

```

1983 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
1984 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh}}
1985 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv}}
1986 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
1987 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
1988 { fitr }
1989 {
1990   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
1991   {
1992     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
1993   }
1994   {
1995     %ensure 4 values ...
1996     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
1997     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1998     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
1999     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2000     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2001     {
2002       \fp_eval:n
2003       {
2004         round
2005         (
2006           abs
2007           (
2008             \seq_item:Nn\l__hyp_tmpa_seq{4}
2009             -
2010             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2011           ),
2012           3
2013         )
2014       }bp
2015     }{}
2016     \box_set_dp:Nn \l__hyp_dest_box
2017     {
2018       \fp_eval:n
2019       {
2020         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2021       }bp
2022     }
2023     \box_set_ht:Nn \l__hyp_dest_box
2024     {
2025       \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2026     }
2027   }
2028 }
2029 {
2030 {
2031   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2032   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2033 }
2034 }
2035 ,pdfview .initial:n = {xyz}
2036 }

```

## 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.

```
2037 \keys_define:nn { hyp / setup }
2038 {
2039   ,pdflang      .code:n =
2040   {
2041     \tl_if_empty:nTF { #1 }
2042     {
2043       \pdfmanagement_remove:nn {Catalog} { Lang }
2044     }
2045     {
2046       \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2047     }
2048     \__hyp_store_metadata:nn {pdflang}{#1}
2049   }
2050 }
```

### 12.3.1 “info keys”

`pdfauthor (setup key)` The keys store their value also in the metadata container, so that hyperxmp can use them.

```
pdfcreator (setup key) 2051 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 2052 {
pdfproducer (setup key) 2053   \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 2054   {
2055     pdf#1 .code:n =
2056     {
2057       \tl_if_blank:nTF {##1}
2058       {
2059         \pdfmanagement_remove:nn {Info}{#2}
2060       }
2061       {
2062         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2063         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2064         {
2065           \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2066         }
2067       }
2068       \__hyp_store_metadata:nn {pdf#1}{##1}
2069     }
2070   }
2071   \keys_define:nn { hyp / info }
2072   {
2073     #2 .code:n =
2074     {
2075       \tl_if_blank:nTF {##1}
2076       {
2077         \pdfmanagement_remove:nn {Info}{#2}
2078       }
2079       {
```

```

2080         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2081         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2082         {
2083             \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2084         }
2085     }
2086     \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2087 }
2088 ,unknown .code:n =
2089 {
2090     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2091     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2092     {
2093         \exp_args:Nno
2094         \pdfmanagement_add:nnx {Info}
2095             { \l_keys_key_str } {\l__hyp_tmpa_str}
2096     }
2097 }
2098 }
2099 }
2100 \__hyp_setup_info_key:nn {author} {Author}
2101 \__hyp_setup_info_key:nn {title} {Title}
2102 \__hyp_setup_info_key:nn {producer} {Producer}
2103 \__hyp_setup_info_key:nn {creator} {Creator}
2104 % ignored key: addtopdfcreator
2105 \__hyp_setup_info_key:nn {subject} {Subject}
2106 \__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.

```

2107 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2108 {
2109     \keys_define:nn { hyp / setup }
2110     {
2111         pdf#1 .code:n =
2112         {
2113             \tl_if_blank:nTF {##1}
2114             {
2115                 \pdfmanagement_remove:nn {Info}{#2}
2116             }
2117             {
2118                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2119             }
2120             \__hyp_store_metadata:nn {pdf#1}{##1}
2121         }
2122     }
2123     \keys_define:nn { hyp / info }
2124     {
2125         #2 .code:n =
2126         {
2127             \tl_if_blank:nTF {##1}
2128             {
2129                 \pdfmanagement_remove:nn {Info}{#2}
2130             }

```

```

2131         {
2132             \pdfmanagement_add:nxx {Info}{#2}{(\tl_to_str:n {##1})}
2133         }
2134         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2135     }
2136 }
2137 }
2138
2139 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2140 \__hyp_setup_info_date_key:nn {moddate} {ModDate}

```

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

```

2141 \keys_define:nn { hyp / setup }
2142 {
2143     ,pdftrapped .code:n =
2144     {
2145         \exp_args:Nne
2146         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2147     }
2148     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2149     {
2150         \pdfmanagement_add:nxx {Info}{Trapped}
2151         {/
2152             \str_uppercase:f { \str_head:n { #1 } }
2153             \str_lowercase:f { \str_tail:n { #1 } }
2154         }
2155         \__hyp_store_metadata:nx {pdftrapped}
2156         {
2157             \str_uppercase:f { \str_head:n { #1 } }
2158             \str_lowercase:f { \str_tail:n { #1 } }
2159         }
2160     }
2161     ,_pdftrapped / unknown .code:n =
2162     {
2163         \msg_warning:nxxx { hyp } { unknown-choice }
2164         { pdftrapped }
2165         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2166         { \exp_not:n {#1} }
2167     }
2168 }

```

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

```

2169 \keys_define:nn { hyp / setup }
2170 {
2171     pdfinfo .code:n =
2172     {
2173         \keys_set:nn { hyp / info } { #1 }
2174     }
2175 }

```

Now we set some default values

```

2176 \keys_set:nn { hyp / setup } {pdfcreator = LaTeX-with-hyperref}
2177 \keys_set:nn { hyp / setup } {pdfauthor = }
2178 \keys_set:nn { hyp / setup } {pdftitle = }
2179 \keys_set:nn { hyp / setup } {pdfsubject = }

```

## 12.4 hyperxmp keys

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

```
2180 \clist_map_inline:nn
2181 {
2182   ,pdfcopyright
2183   ,pdftype
2184   ,pdflicenseurl
2185   ,pdfauthortitle
2186   ,pdfcaptionwriter
2187   ,pdfmetalang
2188   ,pdfapart
2189   ,pdfaconformance
2190   ,pdfuapart
2191   ,pdfxstandard
2192   ,pdfsource
2193   ,pdfdocumentid
2194   ,pdfinstanceid
2195   ,pdfversionid
2196   ,pdfrendition
2197   ,pdfpublication
2198   ,pdfpubtype
2199   ,pdfbytes
2200   ,pdfnumpages
2201   ,pdfissn
2202   ,pdfeissn
2203   ,pdfisbn
2204   ,pdfbookedition
2205   ,pdfpublisher
2206   ,pdfvolumenum
2207   ,pdfissuenum
2208   ,pdfpagerange
2209   ,pdfdoi
2210   ,pdfurl
2211   ,pdfidentifier
2212   ,pdfsubtitle
2213   ,pdfpubstatus
2214   ,pdfcontactaddress
2215   ,pdfcontactcity
2216   ,pdfcontactregion
2217   ,pdfcontactpostcode
2218   ,pdfcontactcountry
2219   ,pdfcontactphone
2220   ,pdfcontactemail
2221   ,pdfcontacturl
2222   ,pdfdate
2223 }
2224 {
2225   \keys_define:nn { hyp / setup }
2226   {
2227     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2228   }
2229 }
```



12.5 Transitions

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

```
2230 \keys_define:nn { hyp / setup }
2231 {
2232   pdfpageduration .code:n =
2233   {
2234     \tl_if_blank:nTF { #1 }
2235     {
2236       \pdfmanagement_remove:nn {Page}{Dur}
2237     }
2238     {
2239       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2240     }
2241   }
2242 }
```

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

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

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

The keys allowed in the argument of `pdfpagetransition` are

```
2243 \keys_define:nn { hyp / setup }
2244 {
2245   pdfpagetransition .code:n =
2246   {
2247     \tl_if_blank:nTF {#1}
2248     {
2249       \pdfmanagement_remove:nn {Page}{Trans}
2250     }
2251     {
```

```

2252         \group_begin:
2253         \keys_set:nn { hyp / trans }{style=R,#1}
2254         \pdf_object_unnamed_write:nx { dict }
2255         {
2256             \pdfdict_use:n {l__hyp_page/Trans}
2257         }
2258         \pdfmanagement_add:nx {Page}{Trans}{\pdf_object_ref_last:}
2259         \group_end:
2260     }
2261 }
2262 }
2263 \keys_define:nn { hyp / trans }
2264 {
2265     ,style .choices:nn =
2266     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2267     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2268     ,style / unknown .code:n =
2269     {
2270         \msg_warning:nnxxx { hyp } { unknown-choice }
2271         { trans / style }
2272         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2273         { \exp_not:n {#1} }
2274     }
2275     ,duration .code:n =
2276     {
2277         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2278     }
2279     ,direction .choices:nn =
2280     {H,V}
2281     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2282     ,direction .choices:nn =
2283     {0,90,180,270,315}
2284     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ #1 } }
2285     ,direction / None .code:n =
2286     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{ /None } }
2287     ,direction / unknown .code:n =
2288     {
2289         \msg_warning:nnxxx { hyp } { unknown-choice }
2290         { trans / direction }
2291         {
2292             H~(horizontal,~only~Split,~Blinds),
2293             V~(vertical,~only~Split,~Blinds),
2294             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2295             90~(bottom~to~top,~only~Wipe),
2296             180~(right~to~left,~only~Wipe),
2297             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2298             315~(top~left~to~bottom,~only~Glitter),
2299             None~(only~Fly)
2300         }
2301         { \exp_not:n {#1} }
2302     }
2303     ,motion .choices:nn =
2304     {I,0}
2305     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }

```

```

2306 ,motion / unknown .code:n =
2307 {
2308     \msg_warning:nnxxx { hyp } { unknown-choice }
2309     { trans / motion }
2310     { I~(inwards) , O~(outwards) }
2311     { \exp_not:n {#1} }
2312 }
2313 ,scale .code:n =
2314 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2315 ,opaque .choices:nn = {true,false}
2316 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2317 ,opaque / unknown .code:n =
2318 {
2319     \msg_warning:nnxxx { hyp } { unknown-choice }
2320     { trans / B }
2321     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2322     { \exp_not:n {#1} }
2323 }
2324 % try to set unknown keys as style
2325 ,unknown .code:n =
2326 {
2327     % warning ...
2328     \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2329 }
2330 }

```

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

```

2331 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2332 \NewDocumentCommand \MakeFieldObject { m m }
2333 {
2334     \pdfxform_new:nnn { #2 }{ } { #1 }
2335 }
2336
2337
2338 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2339 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2340
2341 \let\HyField@afields\ltx@empty
2342 \let\HyField@cofields\ltx@empty
2343 %% UF test for old pdftex removed
2344 %\let\HyField@AuxAddToFields\ltx@gobble
2345 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2346 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2347
2348 % the value doesn't matter, but with a prop we avoid duplicates and it is
2349 % clearly faster than removing them from a sequence
2350 \def\HyField@AuxAddToFields#1
2351 {
2352     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2353 }%
2354
2355 %fields with empty key get a value too -- lets hope that
2356 %this give the expected behaviour

```

```

2357 \def\HyField@AuxAddToCoFields #1 #2
2358 {
2359   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2360 }
2361
2362 \Hy@AtBeginDocument
2363 {
2364   \if@files
2365     \immediate\write\@mainaux{%
2366       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2367     }%
2368     \immediate\write\@mainaux{%
2369       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2370     }%
2371   \fi
2372   \let\HyField@AfterAuxOpen\@firstofone
2373 }%
2374
2375 \def\HyField@AddToFields
2376 {
2377   \exp_args:Nx\HyField__hypAddToFields
2378   {
2379     \pdfannot_box_ref_last:
2380   }
2381   \ifx\Fld@calculate@code\ltx@empty
2382   \else
2383     \begingroup
2384       \Hy@safe@activestrue
2385       \edef\Hy@temp{%
2386         \endgroup
2387         \if@files
2388           \write\@mainaux
2389           {
2390             \string\HyField@AuxAddToCoFields
2391             {
2392               \Fld@calculate@sortkey
2393             }
2394             {
2395               \pdfannot_box_ref_last:
2396             }
2397           }
2398         \fi
2399       }%
2400     \Hy@temp
2401   \fi
2402 }%
2403
2404 \def\HyField__hypAddToFields#1{
2405   \HyField@AfterAuxOpen{%
2406     \if@files
2407       \write\@mainaux{%
2408         \string\HyField@AuxAddToFields{#1}%
2409       }%
2410     \fi

```

```

2411     }%
2412 }%
2413
2414 \ExplSyntaxOff
2415 \ExplSyntaxOn
2416 \tl_new:N \l__hyp_CheckmarkYes_tl
2417 \tl_set:Nn \l__hyp_CheckmarkYes_tl { __hyp_xform_CheckMarkYes }
2418 \tl_new:N \l__hyp_CheckmarkOff_tl
2419 \tl_set:Nn \l__hyp_CheckmarkOff_tl { __hyp_xform_CheckMarkOff }
2420
2421 \def\@Form[#1]
2422 {
2423   \@ifundefined{textcolor}{\let\textcolor\@gobble}{ }
2424   \kvsetkeys{Form}{#1}
2425   \pdf@ifdraftmode{ }
2426   {
2427     \Hy@FormObjects
2428     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2429     {
2430       \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2431       %\pdfmanagement_show:n { Catalog / AcroForm }
2432     }
2433     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2434     {
2435       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2436       {
2437         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2438       }
2439       \seq_sort:Nn \l__hyp_tmpa_seq
2440       {
2441         \int_compare:nNnTF { \pdf@strcmp{##1}{##2} } > { 0 }
2442         { \sort_return_swapped: }
2443         { \sort_return_same: }
2444       }
2445       \seq_map_inline:Nn \l__hyp_tmpa_seq
2446       {
2447         \pdfmanagement_add:nnx { Catalog / AcroForm }
2448         { CO }
2449         {
2450           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2451         }
2452       }
2453     }
2454     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2455     {ZaDb} {\pdf_object_ref:n {l__hyp_font_zapdingbats_obj} }
2456     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2457     {Helv} {\pdf_object_ref:n {l__hyp_font_helvetica_obj} }
2458     \pdfmanagement_add:nnx {Catalog /AcroForm}
2459     {DA}{(/Helv~10~Tf~0~g)}
2460     \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2461     {
2462       \legacy_if:nT { HyField@NeedAppearances }
2463       {
2464         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}

```

```

2465         }
2466     }
2467     {
2468         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2469     }
2470 }
2471 \MakeFieldObject
2472 {
2473     \group_begin:
2474     \fontfamily{pzd}
2475     \fontencoding{U}
2476     \fontseries{m}
2477     \fontshape{n}
2478     \selectfont
2479     \char123
2480     \group_end:
2481 }
2482 {__hyp_xform_Ding}
2483 \MakeFieldObject
2484 {
2485     \group_begin:
2486     \fontfamily{pzd}
2487     \fontencoding{U}
2488     \fontseries{m}
2489     \fontshape{n}
2490     \selectfont
2491     \phantom{\char123}
2492     \group_end:
2493 }
2494 {__hyp_xform_DingOff}
2495 \MakeFieldObject
2496 {
2497     \group_begin:
2498     \fontfamily{pzd}
2499     \fontencoding{U}
2500     \fontseries{m}
2501     \fontshape{n}
2502     \selectfont
2503     \char51
2504     \group_end:
2505 }
2506 {__hyp_xform_CheckMarkYes}
2507 \MakeFieldObject
2508 {
2509     \group_begin:
2510     \fontfamily{pzd}
2511     \fontencoding{U}
2512     \fontseries{m}
2513     \fontshape{n}
2514     \selectfont
2515     \phantom{\char51} %perhaps xetex needs some small glyph ..
2516     \group_end:
2517 }
2518 {__hyp_xform_CheckMarkOff}

```

```

2519 \MakeFieldObject
2520 {
2521   \fbox{\textcolor{yellow}{\textsf{Submit}}}% %color?
2522 }
2523 {__hyp_xform_Submit}
2524 \MakeFieldObject
2525 {
2526   \fbox{\textcolor{yellow}{\textsf{SubmitP}}}% %color?
2527 }
2528 {__hyp_xform_SubmitP}
2529 }
2530 \ExplSyntaxOff
2531 \let\@endForm\ltx@empty
2532 \let\HyAnn@AbsPageLabel\ltx@empty
2533 \let\Fld@pageobjref\ltx@empty
2534
2535 \ExplSyntaxOn
2536 \newcount\HyAnn@Count
2537 \HyAnn@Count=\ltx@zero
2538 \def\HyAnn@AbsPageLabel
2539 {
2540   \global\advance\HyAnn@Count by\ltx@one
2541   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2542   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2543   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2544   \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2545   \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2546 }%
2547 \def\Fld@pageobjref
2548 {
2549   \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2550   {
2551     /P~\pdf_pageobject_ref:n
2552     {
2553       \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2554     }
2555   }
2556 }
2557 \ExplSyntaxOff
2558 \ExplSyntaxOn
2559 %% check if the attr should be set through
2560 %% hooks.
2561 %% check if options are missing.
2562 \def\@TextField[#1]#2{% parameters, label
2563   \def\Fld@name{#2}%
2564   \let\Fld@default\ltx@empty
2565   \let\Fld@value\@empty
2566   \def\Fld@width{\DefaultWidthofText}%
2567   \def\Fld@height{%
2568     \ifFld@multiline
2569       \DefaultHeightofTextMultiline
2570     \else
2571       \DefaultHeightofText
2572     \fi

```

```

2573 }%
2574 \begingroup
2575   \expandafter\HyField@SetKeys\expandafter{%
2576     \DefaultOptionsofText,#1%
2577   }%
2578   \PDFForm@Name
2579   \HyField@FlagsText
2580   \ifFld@hidden\def\Fld@width{1sp}\fi
2581   \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2582   \LayoutTextField{#2}{%
2583     \leavevmode
2584     \HyAnn@AbsPageLabel
2585     \Hy@escapeform\PDFForm@Text
2586     \pdfannot_box:nnnn
2587       {\Fld@width}
2588       {\Fld@height}
2589       {Opt} %is this correct?
2590       {\PDFForm@Text}
2591     \MakeTextField{\Fld@width}{\Fld@height}
2592     \HyField@AddToFields
2593   }%
2594 \endgroup
2595 }
2596 \providecommand\@curropt{}
2597 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2598   \def\Fld@name{#2}
2599   \let\Fld@default\relax
2600   \let\Fld@value\relax
2601   \def\Fld@width{\DefaultWidthofChoiceMenu}
2602   \def\Fld@height{\DefaultHeightofChoiceMenu}
2603   \begingroup
2604     \Fld@menulength=0 %
2605     \@tempdima\z@
2606     \clist_map_variable:nNn { #3 } \@curropt
2607     %\@for\@curropt:=#3\do
2608     {%
2609       \expandafter\Fld@checkequals\@curropt==\%
2610       \Hy@StepCount\Fld@menulength
2611       \settowidth{\@tempdimb}{\@currDisplay}%
2612       \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2613     }%
2614   \advance\@tempdima by~15\p@
2615   \begingroup
2616     \HyField@SetKeys{#1}
2617   \edef\x{\endgroup
2618     \noexpand\expandafter
2619     \noexpand\HyField@SetKeys
2620     \noexpand\expandafter{%
2621       \expandafter\noexpand\csname DefaultOptionsof%
2622         \ifFld@radio
2623           Radio%
2624         \else
2625           \ifFld@combo
2626             \ifFld@popdown

```



```

2627         PopdownBox%
2628     \else
2629         ComboBox%
2630     \fi
2631 \else
2632     ListBox%
2633 \fi
2634 \fi
2635 \endcsname
2636 }%
2637 }\x
2638 \HyField@SetKeys{#1}%
2639 \PDFForm@Name
2640 \ifFld@hidden\def\Fld@width{1sp}\fi
2641 \ifx\Fld@value\relax
2642     \let\Fld@value\Fld@default
2643 \fi
2644 \LayoutChoiceField{#2}{%
2645     \ifFld@radio
2646         \HyField@FlagsRadioButton
2647         \__hypRadio{#3}%
2648     \else
2649         \begingroup
2650         \HyField@FlagsChoice
2651         \ifdim\Fld@width<\@tempdima
2652             \ifdim\@tempdima<1cm\@tempdima1cm\fi
2653             \edef\Fld@width{\the\@tempdima}%
2654         \fi
2655         \ifFld@combo
2656         \else
2657             \@tempdima=\the\Fld@menulength\Fld@charsize
2658             \advance\@tempdima by~\Fld@borderwidth bp %
2659             \advance\@tempdima by~\Fld@borderwidth bp %
2660             \edef\Fld@height{\the\@tempdima}%
2661         \fi
2662         \__hypListbox{#3}%
2663     \endgroup
2664 \fi
2665 }%
2666 \endgroup
2667 }
2668 \tl_new:N \l__hyp_RadioYes_tl
2669 \tl_set:Nn \l__hyp_RadioYes_tl { __hyp_xform_Ding }
2670 \def\__hypRadio#1{%
2671     \Fld@listcount=0-%
2672     %\show\Fld@default
2673     \EdefEscapeName\Fld@default{\Fld@default}%
2674     \clist_map_variable:nNn { #1 } \@curropt
2675     %\@for\@curropt:=#1\do
2676     {%
2677         \expandafter\Fld@checkequals\@curropt==\\%
2678         \EdefEscapeName\@currValue{\@currValue}%
2679         \Hy@StepCount\Fld@listcount
2680         \@currDisplay\space

```

```

2681 \leavevmode
2682 \HyAnn@AbsPageLabel
2683 \Hy@escapeform\PDFForm@Radio
2684 \pdfannot_box:nnnn
2685 {\Fld@width}
2686 {\Fld@height}
2687 {Opt} %is this correct?
2688 {
2689 \PDFForm@Radio
2690 /AP
2691 <<
2692 /N
2693 <<
2694 /\currValue\c_space_tl \pdfxform_ref:o {__hyp_xform_Ding}
2695 %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2696 >>
2697 >>
2698 }
2699 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2700 \int_compare:nNtT { \Fld@listcount} = { 1 }
2701 { \HyField@AddToFields }
2702 \c_space_tl % deliberate space between radio buttons
2703 % to do: --> should be configurable
2704 }%
2705 }
2706
2707 \newcount\Fld@listcount
2708 \def\__hypListbox#1
2709 {
2710 \HyField@PDFChoices{#1}
2711 \mode_leave_vertical:
2712 \HyAnn@AbsPageLabel
2713 \Hy@escapeform\PDFForm@List
2714 \pdf_link_user:nnn
2715 {widget} %perhaps we need more types??
2716 {\PDFForm@List}
2717 {\MakeChoiceField{\Fld@width}{\Fld@height}}
2718 \HyField@AddToFields
2719 }
2720
2721
2722 \def\@PushButton[#1]#2{% parameters, label
2723 \def\Fld@name{#2}%
2724 \group_begin:
2725 \exp_args:No\HyField@SetKeys
2726 {
2727 \DefaultOptionsofPushButton,#1
2728 }
2729 \PDFForm@Name
2730 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2731 {
2732 \HyField@FlagsPushButton
2733 \legacy_if:nT {Fld@hidden}
2734 {

```

```

2735         \def\Fld@width{1sp}
2736     }
2737     \LayoutPushButtonField
2738     {
2739         \mode_leave_vertical:
2740         \HyAnn@AbsPageLabel
2741         \Hy@escapeform\PDFForm@Push
2742         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2743         \pdfannot_box:nnnn
2744         {\box_wd:N\l_tmpa_box}
2745         {\box_ht:N\l_tmpa_box}
2746         {\box_dp:N\l_tmpa_box} %is this correct?
2747         {\PDFForm@Push}
2748         {\box_use:N\l_tmpa_box}
2749         \HyField@AddToFields
2750     }
2751 }
2752 {
2753     \msg_error:nn { hyp }{ pdfa-no-push-button }
2754     \LayoutPushButtonField
2755     {
2756         \mode_leave_vertical:
2757         \MakeButtonField{#2}
2758     }
2759 }
2760 \group_end:
2761 }
2762
2763 \def\@Submit[#1]#2
2764 {
2765     \def\Fld@width {\DefaultWidthofSubmit}
2766     \def\Fld@height{\DefaultHeightofSubmit}
2767     \group_begin:
2768     \exp_args:No\HyField@SetKeys
2769     {
2770         \DefaultOptionsofSubmit,#1
2771     }
2772     \HyField@FlagsPushButton
2773     \HyField@FlagsSubmit
2774     \legacy_if:nT { Fld@hidden }
2775     {
2776         \def\Fld@width{1sp}
2777     }
2778     \mode_leave_vertical:
2779     \HyAnn@AbsPageLabel
2780     \Hy@escapeform\PDFForm@Submit
2781     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2782     \pdfannot_box:nnnn
2783     {\box_wd:N\l_tmpa_box}
2784     {\box_ht:N\l_tmpa_box}
2785     {\box_dp:N\l_tmpa_box} %is this correct?
2786     {
2787         \PDFForm@Submit
2788         /AP<<

```

```

2789         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2790         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2791     >>
2792     }
2793     \HyField@AddToFields
2794     \box_use:N\l_tmpa_box
2795
2796 \group_end:
2797 }
2798
2799 \def\@Reset[#1]#2
2800 {
2801     \def\Fld@width {\DefaultWidthofReset}
2802     \def\Fld@height{\DefaultHeightofReset}
2803     \group_begin:
2804         \exp_args:No\HyField@SetKeys
2805         {
2806             \DefaultOptionsofReset,#1
2807         }
2808     \mode_leave_vertical:
2809     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2810     {
2811         \HyField@FlagsPushButton
2812         \legacy_if:nT { Fld@hidden }
2813         { \def\Fld@width{1sp} }
2814         \HyAnn@AbsPageLabel
2815         \Hy@escapeform\PDFForm@Reset
2816         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2817         \pdfannot_box:nnnn
2818         {\box_wd:N\l_tmpa_box}
2819         {\box_ht:N\l_tmpa_box}
2820         {\box_dp:N\l_tmpa_box} %is this correct?
2821         { \PDFForm@Reset }
2822         \HyField@AddToFields
2823         \box_use:N \l_tmpa_box
2824     }
2825     {
2826         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2827         \MakeButtonField{#2}
2828     }
2829 \group_end:
2830 }
2831
2832 \def\@CheckBox[#1]#2
2833 {% parameters, label
2834     \def\Fld@name{#2}
2835     \def\Fld@default{0}
2836     \group_begin:
2837         \def\Fld@width {\DefaultWidthofCheckBox}
2838         \def\Fld@height{\DefaultHeightofCheckBox}
2839         \exp_args:No\HyField@SetKeys
2840         {
2841             \DefaultOptionsofCheckBox,#1
2842         }

```

```

2843 \PDFForm@Name
2844 \HyField@FlagsCheckBox
2845 \legacy_if:nT { Fld@hidden }
2846 {
2847   \def\Fld@width{1sp}
2848 }
2849 \LayoutCheckField{#2}
2850 {
2851   \mode_leave_vertical:
2852   \HyAnn@AbsPageLabel
2853   \Hy@escapeform\PDFForm@Check
2854   \pdfannot_box:nnnn
2855     {\Fld@width}
2856     {\Fld@height}
2857     {Opt} %is this correct?
2858     {\PDFForm@Check}
2859   \HyField@AddToFields %check if this works with xelatex ...
2860 }
2861 \group_end:
2862 }
2863 \ExplSyntaxOff
2864
2865 %hm. Should a luatex driver use type1 fonts in fields????
2866 \ExplSyntaxOn
2867 \def\Hy@FormObjects
2868 {
2869   \pdf_object_new:nn {l__hyp_encoding_pdffdoc_obj } { dict }
2870   \pdf_object_new:nn {l__hyp_font_zapfdingbats_obj } { dict }
2871   \pdf_object_new:nn {l__hyp_font_helvetica_obj } { dict }
2872   \pdf_object_write:nx {l__hyp_encoding_pdffdoc_obj }
2873   {
2874     /Type/Encoding
2875     /Differences[
2876       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2877       /ring/tilde
2878       \c_space_tl
2879       39/quotesingle
2880       \c_space_tl
2881       96/grave %
2882       \iow_newline:
2883       128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2884       /fraction/guilsinglleft/guilsinglright/minus/perthousand
2885       /quotedblbase/quotedblleft/quotedblright/quoteleft
2886       /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2887       /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2888       \iow_newline:
2889       164/currency
2890       \c_space_tl
2891       166/brokenbar
2892       \c_space_tl
2893       168/dieresis/copyright/ordfeminine
2894       \c_space_tl
2895       172/logicalnot/.notdef/registered/macron/degree/plusminus
2896       /twosuperior/threesuperior/acute/mu

```

```

2897         \c_space_tl
2898         183/periodcentered/cedilla/onesuperior/ordmasculine
2899         \c_space_tl
2900         188/onequarter/onehalf/threequarters
2901         \iow_newline:
2902         192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2903         /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2904         /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2905         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2906         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2907         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2908         /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2909         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2910         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2911         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2912     ]
2913 }
2914 \pdf_object_write:nn {l__hyp_font_zapfdingbats_obj }
2915 {
2916     /Type/Font
2917     /Subtype/Type1
2918     /Name/ZaDb
2919     /BaseFont/ZapfDingbats
2920 }
2921 \pdf_object_write:nx {l__hyp_font_helvetica_obj }
2922 {
2923     /Type/Font
2924     /Subtype/Type1
2925     /Name/Helv
2926     /BaseFont/Helvetica
2927     /Encoding~\pdf_object_ref:n { l__hyp_encoding_pdfdoc_obj }
2928 }
2929 \global\let\Hy@FormObjects\relax
2930 }
2931 \ExplSyntaxOff
2932 \providecommand*{\Fld@pageobjref}{\}
2933 \ifcsname pdf@escapestring\endcsname
2934     \def\Hy@escapeform#1{%
2935         \ifHy@pdfescapeform
2936             \let\Hy@escapestring\pdfescapestring
2937         \else
2938             \let\Hy@escapestring\@firstofone
2939         \fi
2940     }%
2941     \Hy@escapeform{}\}
2942 \else
2943     \let\Hy@escapestring\@firstofone
2944     \def\Hy@escapeform#1{%
2945         \ifHy@pdfescapeform
2946             \def\Hy@escapestring##1{%
2947                 \noexpand\Hy@escapestring{\noexpand##1}%
2948             }%
2949             \edef\Hy@temp{#1}%
2950             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil

```

```

2951     \def\Hy@escapestring##1{%
2952         \ifundefined{Hy@esc@\string##1}{%
2953             ##1%
2954             \ThisShouldNotHappen
2955         }{%
2956             \csname Hy@esc@\string##1\endcsname
2957         }%
2958     }%
2959     \else
2960         \let\Hy@escapestring\@firstofone
2961     \fi
2962 }%
2963 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
2964     \ifx\#3\%
2965     \else
2966         \expandafter
2967         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
2968         \ltx@ReturnAfterFi{%
2969             \Hy__hypescapeform#3\@nil
2970         }%
2971     \fi
2972 }%
2973 \fi
2974 \def\PDFForm@Name{%
2975     \PDFForm__hypName\Fld@name
2976     \ifx\Fld@altname\relax
2977     \else
2978         \PDFForm__hypName\Fld@altname
2979     \fi
2980     \ifx\Fld@mappingname\relax
2981     \else
2982         \PDFForm__hypName\Fld@mappingname
2983     \fi
2984 }
2985 \def\PDFForm__hypName#1{%
2986     \begingroup
2987         \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
2988             \ifHy@unicode
2989                 \Hy@unicodedefalse
2990             \fi
2991         \fi
2992         \pdfstringdef\Hy@gtemp#1%
2993     \endgroup
2994     \let#1\Hy@gtemp
2995 }
2996 \def\Fld@X@additionalactions{%
2997     \ifx\Fld@keystroke@code\@empty
2998     \else
2999         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3000     \fi
3001     \ifx\Fld@format@code\@empty
3002     \else
3003         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3004     \fi

```

```

3005 \ifx\Fld@validate@code\@empty
3006 \else
3007   /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3008 \fi
3009 \ifx\Fld@calculate@code\@empty
3010 \else
3011   /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3012 \fi
3013 \ifx\Fld@onfocus@code\@empty
3014 \else
3015   /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3016 \fi
3017 \ifx\Fld@onblur@code\@empty
3018 \else
3019   /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3020 \fi
3021 \ifx\Fld@onmousedown@code\@empty
3022 \else
3023   /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3024 \fi
3025 \ifx\Fld@onmouseup@code\@empty
3026 \else
3027   /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3028 \fi
3029 \ifx\Fld@onenter@code\@empty
3030 \else
3031   /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3032 \fi
3033 \ifx\Fld@onexit@code\@empty
3034 \else
3035   /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3036 \fi
3037 }
3038 \ExplSyntaxOn
3039 \def\Fld@additionalactions
3040 {%
3041   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{\}
3042   {
3043     \pdfmeta_standard_verify:nT {\annot_widget_no_AA}
3044     {\AA<<\Fld@X@additionalactions>>}
3045   }
3046 }
3047 \ExplSyntaxOff
3048 \def\Fld@annotnames{%
3049   /T(\Fld@name)%
3050   \ifx\Fld@altname\relax
3051   \else
3052     /TU(\Fld@altname)%
3053   \fi
3054   \ifx\Fld@mappingname\relax
3055   \else
3056     /TM(\Fld@mappingname)%
3057   \fi
3058 }

```



```

3059 \ExplSyntaxOn
3060 \def\PDFForm@Check
3061 {
3062   /Subtype/Widget
3063   ~\Fld@annotflags
3064   ~\Fld@pageobjref
3065   ~\Fld@annotnames
3066   /FT/Btn
3067   \Fld@flags
3068   /Q~\Fld@align
3069   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3070   /AP
3071   <<
3072   /N
3073   <<
3074     /Yes~\pdfxform_ref:o{\l__hyp_CheckmarkYes_tl}
3075     /Off~\pdfxform_ref:o{\l__hyp_CheckmarkOff_tl}
3076   >>
3077   >>
3078   /MK<<
3079     \int_compare:nNf {\Fld@rotation}={0}
3080     {
3081       /R~\Fld@rotation
3082     }
3083     \tl_if_empty:NF\Fld@bordercolor
3084     {
3085       /BC[\Fld@bordercolor]
3086     }
3087     \tl_if_empty:NF\Fld@bcolor
3088     {
3089       /BG[\Fld@bcolor]
3090     }
3091     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3092   >>
3093   /DA
3094   (
3095     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3096     \tl_if_empty:NF \Fld@color
3097     {
3098       \c_space_tl \Fld@color
3099     }
3100   )
3101   /H/P
3102   \legacy_if:nTF {\Fld@checked}
3103   {
3104     /V/Yes /AS/Yes
3105   }
3106   {
3107     /V/Off /AS/Off
3108   }
3109   \Fld@additionalactions
3110 }
3111 \ExplSyntaxOff
3112 \ExplSyntaxOn

```

```

3113 \def\PDFForm@Push
3114 {
3115     /Subtype/Widget
3116     ~\Fld@annotflags
3117     ~\Fld@pageobjref
3118     ~\Fld@annotnames
3119     /FT/Btn
3120     ~\Fld@flags
3121     /H/P
3122     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3123     \bool_if:nT
3124     {
3125         !\int_compare:nNn {\Fld@rotation} = {0}
3126         ||
3127         \tl_if_exist_p:N \Fld@bordercolor
3128     }
3129     {
3130         /MK
3131         <<
3132         \int_compare:nNnF {\Fld@rotation} = {0}
3133         {
3134             /R~\Fld@rotation
3135         }
3136         \tl_if_exist:NT \Fld@bordercolor
3137         {
3138             /BC[\Fld@bordercolor]
3139         }
3140         >>
3141     }
3142     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3143     \Fld@additionalactions
3144 }
3145
3146 \ExplSyntaxOff
3147 \def\PDFForm@List{%
3148     /Subtype/Widget%
3149     \Fld@annotflags
3150     \Fld@pageobjref
3151     \Fld@annotnames
3152     /FT/Ch%
3153     \Fld@flags
3154     /Q \Fld@align
3155     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3156     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3157         \ifx\Fld@bordercolor\relax\else 1\fi
3158         \ifx\Fld@bcolor\relax \else 1\fi
3159         \space
3160     \else
3161         /MK<<%
3162         \ifnum\Fld@rotation=\z@
3163         \else
3164             /R \Fld@rotation
3165         \fi
3166         \ifx\Fld@bordercolor\relax

```

```

3167         \else
3168             /BC[\Fld@bordercolor]%
3169         \fi
3170         \ifx\Fld@bcolor\relax
3171         \else
3172             /BG[\Fld@bcolor]%
3173         \fi
3174     >>%
3175 \fi
3176 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3177     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3178 \Fld@choices
3179 \Fld@additionalactions
3180 }
3181 \ExplSyntaxOn
3182 \def\PDFForm@Radio
3183 {
3184     /Subtype/Widget
3185     ~\Fld@annotflags
3186     ~\Fld@pageobjref
3187     ~\Fld@annotnames
3188     /FT/Btn
3189     \Fld@flags
3190     /H/P
3191     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3192     /MK<<
3193     \ifnum\Fld@rotation=\z@
3194     \else
3195         /R~\Fld@rotation
3196     \fi
3197     \ifx\Fld@bordercolor\relax
3198     \else
3199         /BC[\Fld@bordercolor]%
3200     \fi
3201     \ifx\Fld@bcolor\relax
3202     \else
3203         /BG[\Fld@bcolor]%
3204     \fi
3205     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3206     >>
3207     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3208         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3209     \ifx\Fld@default\@empty
3210         /V/Off%
3211         /DV/Off%
3212     \else
3213         /V/\Fld@default
3214         /DV/\Fld@default
3215     \fi
3216     \Fld@additionalactions
3217 }
3218 \ExplSyntaxOff
3219 \ExplSyntaxOn
3220 % Does an appeareance dict make sense here?

```

```

3221 \def\PDFForm@Text
3222 {
3223   /Subtype/Widget
3224   ~\Fld@annotflags
3225   ~\Fld@pageobjref
3226   ~\Fld@annotnames
3227   /FT/Tx
3228   ~\Fld@flags
3229   /Q~\Fld@align
3230   /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3231   \bool_if:nT
3232   {
3233     !\int_compare_p:nNn {\Fld@rotation} = {0}
3234     ||
3235     \tl_if_exist_p:N \Fld@bordercolor
3236     ||
3237     \tl_if_exist_p:N \Fld@bcolor
3238   }
3239   {
3240     /MK
3241     <<
3242       \int_compare:nNf {\Fld@rotation} = {0}
3243       {
3244         /R~\Fld@rotation
3245       }
3246       \tl_if_exist:NT \Fld@bordercolor
3247       {
3248         /BC[\Fld@bordercolor]
3249       }
3250       \tl_if_exist:NT \Fld@bcolor
3251       {
3252         /BG[\Fld@bcolor]
3253       }
3254     >>
3255   }
3256   /DA
3257   (
3258     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3259     \tl_if_empty:NF {\c_space_tl\Fld@color}
3260   )
3261   /DV(\Hy@escapestring{\Fld@default})
3262   /V(\Hy@escapestring{\Fld@value})
3263   ~\Fld@additionalactions
3264   \int_compare:nNt { \Fld@maxlen}>{0}
3265   {
3266     /MaxLen~\Fld@maxlen
3267   }
3268 }
3269 \ExplSyntaxOff
3270
3271 \def\PDFForm@Submit{%
3272   /Subtype/Widget%
3273   \Fld@annotflags
3274   \Fld@pageobjref

```

```

3275 \Fld@annotnames
3276 /FT/Btn%
3277 \Fld@flags
3278 /H/P%
3279 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3280 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3281 \ifx\Fld@bordercolor\relax\else 1\fi
3282 \space
3283 \else
3284 /MK<<%
3285 \ifnum\Fld@rotation=\z@
3286 \else
3287 /R \Fld@rotation
3288 \fi
3289 \ifx\Fld@bordercolor\relax
3290 \else
3291 /BC[\Fld@bordercolor]%
3292 \fi
3293 >>%
3294 \fi
3295 /A<<%
3296 /S/SubmitForm%
3297 /F<<%
3298 /FS/URL%
3299 /F(\Hy@escapestring{\Form@action})%
3300 >>%
3301 \Fld@submitflags
3302 >>%
3303 \Fld@additionalactions
3304 }
3305 \ExplSyntaxOn
3306 \def\PDFForm@Reset{%
3307 /Subtype/Widget%
3308 \Fld@annotflags
3309 \Fld@pageobjref
3310 \Fld@annotnames
3311 /FT/Btn%
3312 \Fld@flags
3313 /H/P%
3314 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3315 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3316 \ifx\Fld@bordercolor\relax\else 1\fi
3317 \space
3318 \else
3319 /MK<<%
3320 \ifnum\Fld@rotation=\z@
3321 \else
3322 /R~\Fld@rotation
3323 \fi
3324 \ifx\Fld@bordercolor\relax
3325 \else
3326 /BC[\Fld@bordercolor]%
3327 \fi
3328 >>%

```

```

3329         \fi
3330         /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3331         /A<</S/ResetForm>>%
3332         \Fld@additionalactions
3333     }%
3334
3335
3336 %these patterns are used in hyperref checks.
3337 %it is unclear if they are really useful and if a backend support is
3338 %needed.
3339 \str_case:VnF \c_sys_backend_str
3340 {
3341     { pdfmode }
3342     {
3343         \def\HyPat@ObjRef
3344         {
3345             [0-9]*[1-9][0-9]*~0~R
3346         }
3347     }
3348     { dvipdfmx }
3349     {
3350         \def\HyPat@ObjRef
3351         {
3352             @[^~]+
3353         }
3354     }
3355     { xdvipdfmx }
3356     {
3357         \def\HyPat@ObjRef
3358         {
3359             @[^~]+
3360         }
3361     }
3362 }
3363 { %also set in hyperref sty, so probably not needed.
3364     \def\HyPat@ObjRef/{.+}
3365 }
3366
3367
3368 \ExplSyntaxOff
3369 % UF: removed Hy@writebookmark
3370 %     \Hy@currentbookmarklevel{0}
3371 %     \Hy@numberline
3372 %     \__hypwritetorep
3373 %     counter{bookmark@seq@number}
3374 % removed \HyPsd@SanitizeForOutFile, not needed
3375 % removed \currentpdfbookmark, defined by bookmark,
3376 % should use \newcommand there
3377 % removed \subpdfbookmark, defined by bookmark,
3378 % should use \newcommand there
3379 % removed \belowpdfbookmark, defined by bookmark,
3380 % should use \newcommand there
3381 % removed \pdfbookmark, defined by bookmark,
3382 % \BOOKMARK

```

```

3383 % \@BOOKMARK
3384 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3385 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3386 %% removed \ReadBookmarks / unneeded with bookmark.
3387 %% removed \Hy@OutlineName
3388 %% removed \check@bm@number
3389 %% removed \calc@bm@number
3390
3391 \ifHy@implicit
3392 \else
3393   \expandafter\endinput
3394 \fi
3395 \newlength\Hy@SectionHShift
3396 \def\Hy@SectionAnchorHref#1{%
3397   \ifx\protect\@typeset@protect
3398     \Hy__hypSectionAnchor{#1}%
3399   \fi
3400 }
3401 \DeclareRobustCommand*\Hy__hypSectionAnchor{[1]}{%
3402   \leavevmode
3403   \hbox to 0pt{%
3404     \kern-\Hy@SectionHShift
3405     \Hy@raisedlink{%
3406       \hyper@anchorstart{#1}\hyper@anchorend
3407     }%
3408     \hss
3409   }%
3410 }
3411 \let\H@old@ssect\@ssect
3412 \def\@ssect#1#2#3#4#5{%
3413   \Hy@MakeCurrentHrefAuto{section*}%
3414   \setlength{\Hy@SectionHShift}{#1}%
3415   \begingroup
3416     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3417     \toks\tw@\expandafter{%
3418       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3419       #5%
3420     }%
3421   \edef\x{\endgroup
3422     \the\toks@{\the\toks\tw@}%
3423   }\x
3424 }
3425 \let\H@old@schapter\@schapter
3426 \def\@schapter#1{%
3427   \begingroup
3428     \let\@mkboth\@gobbletwo
3429     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3430     \Hy@raisedlink{%
3431       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3432     }%
3433   \endgroup
3434   \H@old@schapter{#1}%
3435 }
3436 \ltx@ifundefined{chapter}{-}{%

```

```

3437 \let\Hy@org@chapter\@chapter
3438 \def\@chapter{%
3439   \def\Hy@next{%
3440     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3441     \Hy@raisedlink{%
3442       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3443     }%
3444   }%
3445   \ifnum\c@secnumdepth>\m@ne
3446     \ltx@ifundefined{if@mainmatter}%
3447     \iftrue{\csname if@mainmatter\endcsname}%
3448     \let\Hy@next\relax
3449   \fi
3450   \fi
3451   \Hy@next
3452   \Hy@org@chapter
3453 }%
3454 }
3455 \let\H@old@part\@part
3456 \begingroup\expandafter\expandafter\expandafter\endgroup
3457 \expandafter\ifx\csname chapter\endcsname\relax
3458   \let\Hy@secnum@part\z@
3459 \else
3460   \let\Hy@secnum@part\m@ne
3461 \fi
3462 \def\@part{%
3463   \ifnum\Hy@secnum@part>\c@secnumdepth
3464     \phantomsection
3465   \fi
3466   \H@old@part
3467 }
3468 \let\H@old@spart\@spart
3469 \def\@spart#1{%
3470   \Hy@MakeCurrentHrefAuto{part*}%
3471   \Hy@raisedlink{%
3472     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3473   }%
3474   \H@old@spart{#1}%
3475 }
3476 \let\H@old@sect\@sect
3477 \def\@sect#1#2#3#4#5#6[#7]#8{%
3478   \ifnum #2>\c@secnumdepth
3479     \expandafter\@firstoftwo
3480   \else
3481     \expandafter\@secondoftwo
3482   \fi
3483   {%
3484     \Hy@MakeCurrentHrefAuto{section*}%
3485     \setlength{\Hy@SectionHShift}{#3}%
3486     \begingroup
3487       \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3488       \toks\tw@\expandafter{%
3489         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3490         #8%

```



```

3491     }%
3492     \edef\x{\endgroup
3493       \the\toks@{\the\toks\tw@}%
3494     }\x
3495   }{%
3496     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8}%
3497   }%
3498 }
3499 \expandafter\def\csname Parent-4\endcsname{}
3500 \expandafter\def\csname Parent-3\endcsname{}
3501 \expandafter\def\csname Parent-2\endcsname{}
3502 \expandafter\def\csname Parent-1\endcsname{}
3503 \expandafter\def\csname Parent0\endcsname{}
3504 \expandafter\def\csname Parent1\endcsname{}
3505 \expandafter\def\csname Parent2\endcsname{}
3506 \expandafter\def\csname Parent3\endcsname{}
3507 \expandafter\def\csname Parent4\endcsname{}
3508 %%
3509 %% End of file 'hgeneric-testphase.def'.
3510 \</package>
3511 <*colorscheme>
3512 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3513 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3514 % linkcolor=
3515 %,filecolor=
3516 %,urlcolor=
3517 %,menucolor=
3518 %,runcolor=
3519 %,linkbordercolor=
3520 %,filebordercolor=
3521 %,urlbordercolor=
3522 %,menubordercolor=
3523 %,runbordercolor=
3524
3525 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3526 {
3527   linkcolor      = [rgb]{1,0,0}, %red
3528   filecolor      = [rgb]{0,1,1}, %cyan
3529   urlcolor       = [rgb]{1,0,1}, %magenta
3530   menucolor      = [rgb]{1, 0, 0}, %red
3531   runcolor       = [rgb]{0,1,1}, %cyan
3532   %-----
3533   linkbordercolor = [rgb]{1, 0 ,0 },
3534   filebordercolor = [rgb]{0, .5, .5},
3535   urlbordercolor  = [rgb]{0, 1, 1},
3536   menubordercolor = [rgb]{1, 0, 0},
3537   runbordercolor  = [rgb]{0, .7, .7}
3538 }
3539
3540 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3541 {
3542   linkcolor      = [rgb]{0,0.2,0.6},
3543   filecolor      = [rgb]{0.8,0,0.8},
3544   urlcolor       = [rgb]{0.8,0,0.8},

```

```

3545     menucolor      = [rgb]{0,0.2,0.6},
3546     runcolor       = [rgb]{0.8,0,0.8},
3547 %----- %-----
3548     linkbordercolor = [rgb]{0,0.2,0.6},
3549     filebordercolor = [rgb]{0.8,0,0.8},
3550     urlbordercolor  = [rgb]{0.8,0,0.8},
3551     menubordercolor = [rgb]{0,0.2,0.6},
3552     runbordercolor  = [rgb]{0.8,0,0.8}
3553 }
3554
3555 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3556 { %two colors: intern/extern
3557     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3558     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3559     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3560     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3561     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3562 %----- %-----
3563     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3564     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3565     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3566     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3567     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3568 }
3569
3570 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3571 { %all darkgray
3572     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3573     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3574     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3575     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3576     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3577 %----- %-----
3578     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3579     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3580     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3581     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3582     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3583 }
3584
3585 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3586 { %dvipsnam.def
3587     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3588     filecolor      = [rgb]{1, 0, 0}, %Red
3589     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3590     menucolor      = [rgb]{1, 0, 0}, %Red
3591     runcolor       = [rgb]{1, 0, 0}, %Red
3592 %----- %-----
3593     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3594     filebordercolor = [rgb]{1, 0, 0}, %Red
3595     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3596     menubordercolor = [rgb]{1, 0, 0}, %Red
3597     runbordercolor  = [rgb]{1, 0, 0} %Red
3598 }

```

```

3599
3600 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3601 { %dvipsnam.def
3602   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3603   filecolor      = [rgb]{0, 1, 0},    %Green
3604   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3605   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3606   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3607   %----- %-----
3608   linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3609   filebordercolor = [rgb]{0, 1, 0},    %Green
3610   urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3611   menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3612   runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3613 }
3614
3615
3616 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3617 {
3618   linkcolor      = [rgb]{0.50196, 0, 0.02353},
3619   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3620   urlcolor       = [rgb]{0.54118, 0, 0.52941},
3621   menucolor      = [rgb]{0.44706, 0.45882, 0},
3622   runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3623   %----- %-----
3624   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3625   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3626   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3627   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3628   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3629 }
3630
3631 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3632 {
3633   linkcolor      = [rgb]{0,0,0},
3634   filecolor      = [rgb]{0,0,0},
3635   urlcolor       = [rgb]{0,0,0},
3636   menucolor      = [rgb]{0,0,0},
3637   runcolor       = [rgb]{0,0,0},
3638   %----- %-----
3639   linkbordercolor = [rgb]{0,0,0},
3640   filebordercolor = [rgb]{0,0,0},
3641   urlbordercolor  = [rgb]{0,0,0},
3642   menubordercolor = [rgb]{0,0,0},
3643   runbordercolor  = [rgb]{0,0,0}
3644 }
3645 </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	
\#	637
\%	638
\.	455, 459, 461
@curropt commands:	
\@curropt:	2607, 2675
\\	17, 18, 19, 26, 30, 40, 60, 67, 74, 81, 88, 103, 108, 109, 117, 118, 127, 128, 129, 130, 131, 138, 767, 2609, 2677, 2964
\_	453, 455, 459, 461
A	
\A	453
\Acrobatmenu	17, 155
\addcontentsline	12
\AddToDocumentProperties	329
\AddToHook	354
\advance	2540, 2614, 2658, 2659
allcolor (hypersetup key)	896
\author	2
B	
\b	457
\begingroup	183, 237, 273, 2383, 2574, 2603, 2615, 2649, 2986, 3415, 3427, 3456, 3486
\belowpdfbookmark	3379
\bgroup	247
\BOOKMARK	3382
bookmarkstype (hypersetup key)	12
bool commands:	
\bool_if:N	240, 255, 303, 586, 601, 612, 625, 633, 663, 721, 778, 866, 876, 1116, 1127
\bool_if:N	733, 1819, 1844, 3123, 3231
\bool_lazy_and:nn	349
\bool_new:N	177, 178, 439, 443, 447
\bool_set_true:N	448
bordercolormodel (hypersetup key)	12, 917
box commands:	
\box_dp:N	529, 2746, 2785, 2820
\box_ht:N	528, 2745, 2784, 2819
\box_new:N	388, 450
\box_set_dp:Nn	2016
\box_set_ht:Nn	2023
\box_use:N	1133, 2748, 2794, 2823
\box_use_drop:N	1138
\box_wd:N	527, 2744, 2783, 2818
\l_tmpa_box	2742, 2744, 2745, 2746, 2748, 2781, 2783, 2784, 2785, 2794, 2816, 2818, 2819, 2820, 2823
C	
\char	2479, 2491, 2503, 2515
\chardef	136
\cite	30
clist commands:	
\clist_map_function:nN	110, 119
\clist_map_inline:nn	2180
\clist_map_variable:nNn	2606, 2674
color commands:	
\color_export:nnN	816
\color_export:nnnN	822
\color_select:n	833, 869, 1137
\color_select:nn	839
\color_set:nn	2, 334, 850
\color_set:nnn	2, 333, 856
color names:	
hyp/annot/file	436
hyp/annot/link	436
hyp/annot/menu	436
hyp/annot/run	436
hyp/annot/url	436
colorfile (hypersetup key)	896
colorlink (hypersetup key)	896
colorlinks (hypersetup key)	882
colormenu (hypersetup key)	896
colorrun (hypersetup key)	896
colorscheme (hypersetup key)	1, 1300
colorurl (hypersetup key)	896
cs commands:	
\cs_generate_variant:Nn	134, 135, 331, 367, 382, 387, 491, 825, 842, 859
\cs_gset:Npn	1108
\cs_if_exist_p:N	350
\cs_new:Npn	103, 383
\cs_new_protected:Npn	183, 264, 279, 292, 313, 326, 361, 370, 467, 471, 475, 479, 492, 494, 498, 540, 546, 553, 557, 579, 584, 599, 610, 623, 661, 719, 776, 809, 820, 826, 837, 843, 854, 1052, 2051, 2107
\cs_set_eq:NN	139, 148, 149, 189, 231, 637, 638, 1377

<code>\cs_set_protected:Npn</code> .....	137, 162, 508, 767	<code>\else</code> .....	2382, 2570, 2624, 2628, 2631, 2648, 2656, 2937, 2942, 2959, 2965, 2977, 2981, 2998, 3002, 3006, 3010, 3014, 3018, 3022, 3026, 3030, 3034, 3051, 3055, 3156, 3157, 3158, 3160, 3163, 3167, 3171, 3177, 3194, 3198, 3202, 3208, 3212, 3280, 3281, 3283, 3286, 3290, 3315, 3316, 3318, 3321, 3325, 3392, 3459, 3480
<code>\csname</code> .....	2621, 2956, 2967, 3447, 3457, 3499, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507	<code>\endcsname</code> .....	2635, 2933, 2956, 2967, 3447, 3457, 3499, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507
<code>\currentpdfbookmark</code> .....	3375	<code>\endgroup</code> ..	281, 2386, 2594, 2617, 2663, 2666, 2993, 3421, 3433, 3456, 3492
<b>D</b>			
<code>\d</code> .....	455, 459, 461	<code>\endinginput</code> .....	3393
<code>debug</code> (hypersetup key) .....	<u>1324</u>	exp commands:	
<code>\DeclareDocumentMetadata</code> .....	3, 10, 13, 14, 17, 19, 27, 33	<code>\exp_args:Ne</code> .....	3041
<code>\DeclareRobustCommand</code> .....	236, 251, 271, 284, 298, 3401	<code>\exp_args:Nne</code> .....	2145
<code>\def</code> .....	143, 503, 2346, 2350, 2357, 2375, 2404, 2421, 2538, 2547, 2562, 2563, 2566, 2567, 2580, 2581, 2597, 2598, 2601, 2602, 2640, 2670, 2708, 2722, 2723, 2735, 2763, 2765, 2766, 2776, 2799, 2801, 2802, 2813, 2832, 2834, 2835, 2837, 2838, 2847, 2867, 2934, 2944, 2946, 2951, 2963, 2974, 2985, 2996, 3039, 3048, 3060, 3113, 3147, 3182, 3221, 3271, 3306, 3343, 3350, 3357, 3364, 3396, 3412, 3426, 3438, 3439, 3462, 3469, 3477, 3499, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507	<code>\exp_args:Nno</code> .....	281, 294
<code>\DefaultHeightofCheckBox</code> .....	2838	<code>\exp_args:Nno</code> .....	266, 315, 2093
<code>\DefaultHeightofChoiceMenu</code> .....	2602	<code>\exp_args:Nnox</code> .....	671
<code>\DefaultHeightofReset</code> .....	2802	<code>\exp_args:NNV</code> .....	1804, 1835
<code>\DefaultHeightofSubmit</code> .....	2766	<code>\exp_args:NnV</code> .....	542, 549
<code>\DefaultHeightofText</code> .....	2571	<code>\exp_args:Nnx</code> .....	2328
<code>\DefaultHeightofTextMultiline</code> ...	2569	<code>\exp_args:No</code> .....	571, 575, 2725, 2768, 2804, 2839
<code>\DefaultOptionsofCheckBox</code> .....	2841	<code>\exp_args:NV</code> .....	525, 532
<code>\DefaultOptionsofPushButton</code> .....	2727	<code>\exp_args:Nx</code> ...	667, 2086, 2134, 2377
<code>\DefaultOptionsofReset</code> .....	2806	<code>\exp_not:N</code> .....	767
<code>\DefaultOptionsofSubmit</code> .....	2770	<code>\exp_not:n</code> .....	1239, 1273, 1384, 1444, 1465, 1502, 1522, 1545, 1589, 1622, 1647, 1698, 1723, 1750, 1799, 1876, 1904, 1931, 1951, 2166, 2273, 2301, 2311, 2322
<code>\DefaultOptionsofText</code> .....	2576	<code>\expandafter</code> .....	2575, 2609, 2618, 2620, 2621, 2677, 2950, 2966, 3393, 3417, 3418, 3456, 3457, 3479, 3481, 3488, 3489, 3499, 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507
<code>\DefaultWidthofCheckBox</code> .....	2837	<code>\ExplSyntaxOff</code> .....	9, 2414, 2530, 2557, 2863, 2931, 3047, 3111, 3146, 3218, 3269, 3368
<code>\DefaultWidthofChoiceMenu</code> .....	2601	<code>\ExplSyntaxOn</code> .....	7, 10, 2415, 2535, 2558, 2866, 3038, 3059, 3112, 3181, 3219, 3305
<code>\DefaultWidthofReset</code> .....	2801	<code>extension</code> (hypersetup key) .....	<u>12</u> , <u>1346</u>
<code>\DefaultWidthofSubmit</code> .....	2765	<b>F</b>	
<code>\DefaultWidthofText</code> .....	2566	<code>\fbox</code> .....	2521, 2526, 2699
<code>destlabel</code> (hypersetup key) .....	<u>12</u>	<code>\fi</code> .....	2371, 2398, 2401, 2410, 2572, 2580, 2581, 2612, 2630, 2633, 2634, 2640, 2643, 2652, 2654, 2661, 2664, 2939, 2961, 2971, 2973, 2979, 2983, 2990, 2991,
dim commands:			
<code>\dim_to_decimal_in_bp:n</code> .....	11, 139		
<code>\do</code> .....	2607, 2675		
<code>\documentclass</code> .....	20		
<code>draft</code> (hypersetup key) .....	<u>1324</u>		
<b>E</b>			
<code>\edef</code> 2385, 2617, 2653, 2660, 2949, 3421, 3492			
<code>\EdefEscapeName</code> .....	2673, 2678		



\__hyp_colormodel_select_aux:wn .	\l__hyp_href_url_protocol_tl . . .
..... 826, 830, 837	..... 179, 190, 229, 266, 317
\__hyp_colormodel_set_aux:nwn . . .	\__hyp_link_goto_begin:nw . . . . .
..... 843, 847, 854	..... 557, 591, 606
\c__hyp_colorscheme_daleif_prop 3540	\__hyp_link_goto_end: . . 579, 592, 614
\c__hyp_colorscheme_henryford_-	\c__hyp_map_annot_hyp_prop . . . . 400
prop . . . . . 3631	\c__hyp_map_hyp_annot_prop . . 400,
\c__hyp_colorscheme_julian_prop 3555	860, 924, 960, 987, 995, 1006, 1033,
\c__hyp_colorscheme_original_-	1041, 1110, 1214, 1250, 1260, 1392
prop . . . . . 3525	\__hyp_ocg_init: 1052, 1052, 1108, 1118
\c__hyp_colorscheme_phelype_prop	\__hyp_PageLabels_gpsh: 498, 498, 511
..... 3616	\l__hyp_para_tpa_tl 396, 732, 735, 744
\c__hyp_colorscheme_szabolcsA_-	\l__hyp_para_tpa_tl_uuu\l__hyp_-
prop . . . . . 3585	text_tpa_str_uuu\g__hyp_text_-
\c__hyp_colorscheme_szabolcsB_-	tpa_str . . . . . 393
prop . . . . . 3600	\l__hyp_RadioYes_tl . . . . 2668, 2669
\c__hyp_colorscheme_tivv_prop . . 3570	\__hyp_ref_check:nn 360, 370, 382, 2545
\l__hyp_dest_box . . . . . 29,	\__hyp_ref_if_exist:nn . 360, 368, 369
450, 527, 528, 529, 2000, 2016, 2023	\__hyp_ref_if_exist:nnTF . . 372, 2549
\l__hyp_dest_name_tpa_tl . . 393,	\__hyp_ref_label:nn 360, 361, 367, 2544
560, 561, 567, 571, 573, 576, 684, 697	\__hyp_ref_value:nn 360, 383, 387, 2553
\l__hyp_dest_pdfremotestartview_-	\__hyp_secondoftwoewithopt:wnn . . .
tl . . . . . 419, 692, 1807, 1811	..... 322, 323, 324, 325
\g__hyp_dest_pdfstartpage_tl . . .	\__hyp_setup_info_date_key:nn . . .
..... 419, 1818, 1820, 1827, 1845, 1852	..... 2107, 2139, 2140
\g__hyp_dest_pdfstartview_tl . . .	\__hyp_setup_info_key:nn . . 2051,
..... 419, 1820, 1827, 1838, 1842, 1845, 1852	2100, 2101, 2102, 2103, 2105, 2106
\l__hyp_dest_pdfview_tl . . . . .	\__hyp_store_metadata:nn . . . . .
..... 435, 543, 550,	..... 326, 331, 2048,
1969, 1972, 1979, 1982, 1983, 1984,	2068, 2086, 2120, 2134, 2155, 2227
1985, 1986, 1987, 1992, 1996, 2032	\__hyp_text_cleanup:N . 471, 471, 485
\c__hyp_dest_startview_regex . . .	\l__hyp_text_enc_dest_print_tl . .
..... 451, 1805, 1836	..... 422, 683
\c__hyp_dest_undefined_tl . . . . .	\l__hyp_text_enc_dest_tl . . . . .
..... 399, 566, 567	..... 29, 422, 521, 572
\__hyp_destination:nn . . . . .	\l__hyp_text_enc_file_print_tl . .
..... 29, 514, 514, 543, 550	..... 422, 726
\l__hyp_filename_tpa_tl . . . 393,	\l__hyp_text_enc_info_print_tl . .
666, 668, 674, 675, 680, 727, 728, 744	..... 422, 496
\__hyp_href_pdf_aux:nn . . . . 276, 279	\l__hyp_text_enc_para_print_tl . .
\l__hyp_href_pdf_destination_tl . .	..... 422, 731
..... 180, 191, 230, 281	\l__hyp_text_enc_uri_print_tl . . .
\l__hyp_href_pdf_page_tl 181, 199, 691	..... 242,
\__hyp_href_run_aux:nn . . . . 289, 292	245, 257, 260, 305, 308, 422, 630, 1403
\l__hyp_href_run_parameter_tl . . .	\__hyp_text_pdfstring:nnN . . . . .
..... 182, 203, 294	..... 479, 479, 491, 493,
\__hyp_href_url_aux:n . . . . 310, 313	496, 519, 570, 628, 681, 724, 729, 1403
\__hyp_href_url_aux:nn . . . . 262, 264	\__hyp_text_pdfstring_info:nN . . .
\l__hyp_href_url_encode_bool . . .	..... 494, 494, 2062, 2080, 2090
..... 177, 188, 226, 240, 255, 303	\__hyp_text_purify:nN . 467, 467, 484
\__hyp_href_url_format . . . . .	\__hyp_text_string_from_unicode:nN
..... 183, 189, 231, 316	..... 475, 475, 486
\l__hyp_href_url_ismap_bool . . . .	\g__hyp_text_tpa_str . . 398, 487, 489
..... 178, 202, 633	

\l__hyp_text_tmpa_str .....	linkfileprefix .....	1346
..... 397, 484, 485, 486, 487	linktoc .....	1346
\l__hyp_tmpa_box 388, 1120, 1133, 1138	linktocpage .....	1346
\l__hyp_tmpa_int .....	localanchorname .....	1346
..... 388	menu .....	9, 1392
\l__hyp_tmpa_seq .....	menuborder .....	13
1807, 1836, 1838, 1959, 1960, 1964,	menuborderstyle .....	13, 1006
1966, 1990, 1997, 1998, 1999, 2008,	menucolor .....	896
2010, 2020, 2025, 2437, 2439, 2445	naturalnames .....	1346
\l__hyp_tmpa_str .....	nesting .....	13
..... 388, 2062, 2063, 2065,	ocgcolorfile .....	1146
2080, 2081, 2083, 2090, 2091, 2095	ocgcolorlink .....	1146
\l__hyp_tmpa_tl 388, 522, 526, 534,	ocgcolorlinks .....	1146
937, 941, 1403, 1404, 1409, 1803,	ocgcolormenu .....	1146
1805, 1834, 1836, 1966, 1967, 1974	ocgcolorrun .....	1146
\l__hyp_uri_tmpa_tl ... 393, 631, 632	ocgcolorurl .....	1146
hyp/annot/file (color name) .....	pageanchor .....	1346
..... 436	pdfauthor .....	2051
hyp/annot/link (color name) .....	pdfborder .....	13
..... 436	pdfborderstyle .....	13, 1006
hyp/annot/menu (color name) .....	pdfcreationdate .....	13, 2107
..... 436	pdfcreator .....	2051
hyp/annot/run (color name) .....	pdfencoding .....	1314
..... 436	pdfinfo .....	2169
hyp/annot/url (color name) .....	pdfkeywords .....	2051
..... 436	pdflang .....	13, 2037
hyp/text/pdfstring .....	pdflinkmargin .....	13
..... 466	pdfmoddate .....	13, 2107
\hypercalcbp .....	pdfproducer .....	2051
..... 11, 17, 139	pdfremotestartview .....	11
\HyperDestNameFilter .....	pdfstartview .....	11
..... 520, 571	pdfsubject .....	2051
\hypersetup 1, 2, 5, 8, 9, 12, 17, 64, 96, 162	pdftitle .....	2051
\hypersetup keys:	pdftrapped .....	2141
allcolor .....	pdfversion .....	1314
..... 896	pdfview .....	11, 1955
bookmarkstyle .....	plainpages .....	1346
..... 12	run .....	9, 1392
bordercolormodel .....	runborder .....	13
..... 12, 917	runborderstyle .....	13, 1006
colorfile .....	runcolor .....	896
..... 896	unicode .....	1314
colorlink .....	url .....	9, 1392
..... 896	urlborder .....	13
colorlinks .....	urlborderstyle .....	13, 1006
..... 882	urlcolor .....	896
colormenu .....	verbose .....	1324
..... 896	hypertextnames (hypersetup key) ....	1346
colorrun .....		
..... 896		
colorscheme .....		
..... 1, 1300		
colorurl .....		
..... 896		
debug .....		
..... 1324		
destlabel .....		
..... 12		
draft .....		
..... 1324		
extension .....		
..... 12, 1346		
file .....		
..... 9, 1392		
fileborderstyle .....		
..... 13, 1006		
filecolor .....		
..... 896		
final .....		
..... 1324		
hidefile .....		
..... 1276		
hidelink .....		
..... 1276		
hidelinks .....		
..... 1276		
hidemenu .....		
..... 1276		
hiderun .....		
..... 1276		
hideurl .....		
..... 1276		
hypertextnames .....		
..... 1346		
link .....		
..... 9, 1392		
linkborder .....		
..... 13		
linkborderstyle .....		
..... 13, 1006		
linkcolor .....		
..... 896		

## I

\ifcase .....	3156, 3280, 3315
\ifcsname .....	2933
\ifdim .....	2612, 2651, 2652



`\ifnum` . . . 2987, 3156, 3162, 3193, 3280,  
3285, 3315, 3320, 3445, 3463, 3478  
`\iftrue` . . . . . 3447  
`\ifx` . . . . . 2381, 2581, 2641,  
2964, 2976, 2980, 2997, 3001, 3005,  
3009, 3013, 3017, 3021, 3025, 3029,  
3033, 3050, 3054, 3157, 3158, 3166,  
3170, 3177, 3197, 3201, 3208, 3209,  
3281, 3289, 3316, 3324, 3397, 3457  
`\immediate` . . . . . 2365, 2368  
int commands:  
`\int_compare:nNnTF` . . . 1964, 1990,  
2441, 2700, 3079, 3132, 3242, 3264  
`\int_compare_p:nNn` . . . . 3125, 3233  
`\int_eval:n` . . . . . 690  
`\int_max:nn` . . . . . 691  
`\int_new:N` . . . . . 391  
int internal commands:  
`\__hypListbox` . . . . . 2662, 2708  
`\__hypRadio` . . . . . 2647, 2670  
`\__hypwritetorep` . . . . . 3372  
iow commands:  
`\iow_newline:` . . . . . 2882, 2888, 2901  
  

### K

`\kern` . . . . . 3404  
keys commands:  
`\keys_define:nn` . . . . .  
. . . . 168, 186, 224, 335, 882, 898,  
905, 917, 926, 947, 962, 981, 1008,  
1027, 1148, 1160, 1174, 1195, 1208,  
1216, 1245, 1276, 1289, 1300, 1314,  
1324, 1333, 1346, 1373, 1394, 1399,  
1955, 2037, 2053, 2071, 2109, 2123,  
2141, 2169, 2225, 2230, 2243, 2263  
`\l_keys_key_str` . . . . . 173, 2095, 2328  
`\keys_set:nn` . . . . .  
. 165, 194, 213, 238, 253, 274, 287,  
301, 343, 1306, 1313, 2146, 2173,  
2176, 2177, 2178, 2179, 2253, 2328  
`\keys_set_known:nn` . . . . . 2331  
`\kvsetkeys` . . . . . 164, 2424  
  

### L

`\label` . . . . . 8, 12  
`\LayoutCheckField` . . . . . 2849  
`\LayoutChoiceField` . . . . . 2644  
`\LayoutPushButtonField` . . . . . 2737, 2754  
`\LayoutTextField` . . . . . 2582  
`\leavevmode` . . . . . 2583, 2681, 3402  
legacy commands:  
`\legacy_if:nTF` . . . . 150, 352, 356,  
506, 2462, 2733, 2774, 2812, 2845, 3102  
  
`\let` . . . . . 146, 239, 254,  
275, 288, 302, 323, 324, 325, 596,  
646, 653, 707, 713, 757, 763, 806,  
2341, 2342, 2344, 2345, 2372, 2423,  
2531, 2532, 2533, 2564, 2565, 2599,  
2600, 2642, 2929, 2936, 2938, 2943,  
2960, 2994, 3411, 3425, 3428, 3437,  
3448, 3455, 3458, 3460, 3468, 3476  
`link` (hypersetup key) . . . . . 9, 1392  
`linkborder` (hypersetup key) . . . . . 13  
`linkborderstyle` (hypersetup key) . . . . . 13, 1006  
`linkcolor` (hypersetup key) . . . . . 896  
`linkfileprefix` (hypersetup key) . . . . . 1346  
`linktoc` (hypersetup key) . . . . . 1346  
`linktocpage` (hypersetup key) . . . . . 1346  
`localanchorname` (hypersetup key) . . . . . 1346  
  

### M

`\MakeButtonField` . . . . .  
. . . . . 2742, 2757, 2781, 2816, 2827  
`\MakeChoiceField` . . . . . 2717  
`\MakeFieldObject` . . . . . 2332,  
2471, 2483, 2495, 2507, 2519, 2524  
`\MakeRadioField` . . . . . 2699  
`\MakeTextField` . . . . . 2591  
`\mbox` . . . . . 1130  
`menu` (hypersetup key) . . . . . 9, 1392  
`menuborder` (hypersetup key) . . . . . 13  
`menuborderstyle` (hypersetup key) . . . . . 13, 1006  
`menucolor` (hypersetup key) . . . . . 896  
mode commands:  
`\mode_if_horizontal:TF` . . . . . 516, 538  
`\mode_leave_vertical:` . . . . .  
. . . . . 559, 640, 699, 746,  
783, 2711, 2739, 2756, 2778, 2808, 2851  
msg commands:  
`\msg_error:nn` . . . . . 351, 2753, 2826  
`\msg_line_context:` . . . . . 68  
`\g_msg_module_name_prop` . . . . . 11  
`\msg_new:nnn` . . . . . 36, 43, 48, 52,  
56, 63, 70, 77, 84, 91, 97, 104, 113, 123  
`\msg_new:nnnn` . . . . . 12, 22  
`\msg_warning:nn` . . . . . 152, 1320  
`\msg_warning:nnn` . . . . . 161, 563, 801  
`\msg_warning:nnnn` . . . . .  
. . . . . 1152, 1199, 1442, 1486,  
1520, 1543, 1605, 1634, 1668, 1683,  
1696, 1709, 1734, 1767, 1783, 1810,  
1841, 1874, 1888, 1915, 1949, 2031  
`\msg_warning:nnnnn` . . . . . 172,  
1236, 1270, 1381, 1462, 1499, 1586,  
1619, 1644, 1720, 1747, 1796, 1901,  
1928, 2163, 2270, 2289, 2308, 2319

<b>N</b>	
naturalnames (hypersetup key) . . . . .	<a href="#">1346</a>
nesting (hypersetup key) . . . . .	<a href="#">13</a>
\newcommand . . . . .	<a href="#">142</a> , <a href="#">3376</a> , <a href="#">3378</a> , <a href="#">3380</a>
\newcount . . . . .	<a href="#">2536</a> , <a href="#">2707</a>
\NewDocumentCommand . . . . .	<a href="#">2332</a>
\NewExpandableDocumentCommand . . . . .	<a href="#">322</a>
\newlength . . . . .	<a href="#">3395</a>
\noexpand . . . . .	<a href="#">2618</a> , <a href="#">2619</a> , <a href="#">2620</a> , <a href="#">2621</a> , <a href="#">2947</a>
\nolinkurl . . . . .	<a href="#">3</a> , <a href="#">4</a>
<b>O</b>	
ocgcolorfile (hypersetup key) . . . . .	<a href="#">1146</a>
ocgcolorlink (hypersetup key) . . . . .	<a href="#">1146</a>
ocgcolorlinks (hypersetup key) . . . . .	<a href="#">1146</a>
ocgcolormenu (hypersetup key) . . . . .	<a href="#">1146</a>
ocgcolorrun (hypersetup key) . . . . .	<a href="#">1146</a>
ocgcolorurl (hypersetup key) . . . . .	<a href="#">1146</a>
<b>P</b>	
page/Trans internal commands:	
l_hyp_page/Trans . . . . .	<a href="#">464</a>
pageanchor (hypersetup key) . . . . .	<a href="#">1346</a>
\paperwidth . . . . .	<a href="#">3</a>
\PassOptionsToPackage . . . . .	<a href="#">358</a> , <a href="#">1338</a> , <a href="#">1343</a>
pdf commands:	
\pdf_bdcobject:nn . . . . .	<a href="#">1132</a> , <a href="#">1135</a>
\pdf_destination:nn . . . . .	<a href="#">24</a> , <a href="#">134</a> , <a href="#">533</a>
\pdf_destination:nnnn . . . . .	<a href="#">526</a>
\pdf_emc: . . . . .	<a href="#">1134</a> , <a href="#">1140</a>
\pdf_link_user:nnn . . . . .	<a href="#">2714</a>
\pdf_name_from_unicode_e:n . . . . .	<a href="#">219</a> , <a href="#">670</a> , <a href="#">785</a>
\pdf_object_if_exist:nTF . . . . .	<a href="#">668</a>
\pdf_object_new:nn . . . . .	<a href="#">1054</a> , <a href="#">1055</a> , <a href="#">1056</a> , <a href="#">1057</a> , <a href="#">2869</a> , <a href="#">2870</a> , <a href="#">2871</a>
\pdf_object_ref:n . . . . .	<a href="#">135</a> , <a href="#">680</a> , <a href="#">1060</a> , <a href="#">1062</a> , <a href="#">1084</a> , <a href="#">1085</a> , <a href="#">1088</a> , <a href="#">1092</a> , <a href="#">1097</a> , <a href="#">1102</a> , <a href="#">1107</a> , <a href="#">2455</a> , <a href="#">2457</a> , <a href="#">2927</a>
\pdf_object_ref_last: . . . . .	<a href="#">701</a> , <a href="#">2258</a>
\pdf_object_unnamed_write:nn . . . . .	<a href="#">700</a> , <a href="#">2254</a>
\pdf_object_write:nn . . . . .	<a href="#">1058</a> , <a href="#">1064</a> , <a href="#">1074</a> , <a href="#">1086</a> , <a href="#">2872</a> , <a href="#">2914</a> , <a href="#">2921</a>
\pdf_pageobject_ref:n . . . . .	<a href="#">1827</a> , <a href="#">1852</a> , <a href="#">2551</a>
\pdf_string_from_unicode:nnN . . . . .	<a href="#">477</a>
\pdf_version: . . . . .	<a href="#">1490</a> , <a href="#">1609</a> , <a href="#">1638</a> , <a href="#">1672</a> , <a href="#">1687</a> , <a href="#">1713</a> , <a href="#">1738</a> , <a href="#">1771</a> , <a href="#">1787</a> , <a href="#">1892</a> , <a href="#">1919</a>
\pdf_version_compare:NnTF . . . . .	<a href="#">1146</a> , <a href="#">1193</a> , <a href="#">1480</a> , <a href="#">1571</a> , <a href="#">1593</a> , <a href="#">1629</a> , <a href="#">1662</a> , <a href="#">1677</a> , <a href="#">1703</a> , <a href="#">1728</a> , <a href="#">1754</a> , <a href="#">1777</a> , <a href="#">1882</a> , <a href="#">1909</a>
\pdf_version_compare_p:Nn . . . . .	<a href="#">737</a>
\pdf_version_major: . . . . .	<a href="#">149</a> , <a href="#">1155</a> , <a href="#">1203</a>
\pdf_version_minor: . . . . .	<a href="#">148</a> , <a href="#">1155</a> , <a href="#">1203</a>
pdfannot commands:	
\pdfannot_box:nnnn . . . . .	<a href="#">2586</a> , <a href="#">2684</a> , <a href="#">2743</a> , <a href="#">2782</a> , <a href="#">2817</a> , <a href="#">2854</a>
\pdfannot_box_ref_last: . . . . .	<a href="#">2379</a> , <a href="#">2395</a>
\pdfannot_dict_put:nnn . . . . .	<a href="#">641</a> , <a href="#">701</a> , <a href="#">747</a> , <a href="#">786</a> , <a href="#">938</a> , <a href="#">973</a> , <a href="#">997</a> , <a href="#">1019</a> , <a href="#">1043</a> , <a href="#">1221</a> , <a href="#">1252</a>
\pdfannot_dict_remove:nn . . . . .	<a href="#">932</a> , <a href="#">968</a> , <a href="#">989</a> , <a href="#">1014</a> , <a href="#">1035</a> , <a href="#">1229</a> , <a href="#">1262</a>
\pdfannot_link:nnn . . . . .	<a href="#">642</a> , <a href="#">702</a> , <a href="#">748</a> , <a href="#">787</a>
\pdfannot_link_goto_begin:nw . . . . .	<a href="#">576</a>
\pdfannot_link_goto_end: . . . . .	<a href="#">581</a>
\pdfannot_link_margin:n . . . . .	<a href="#">8</a> , <a href="#">1524</a>
\c_pdfannot_link_types_seq . . . . .	<a href="#">445</a>
pdfauthor (hypersetup key) . . . . .	<a href="#">2051</a>
\pdfbookmark . . . . .	<a href="#">3381</a>
pdfborder (hypersetup key) . . . . .	<a href="#">13</a>
pdfborderstyle (hypersetup key) . . . . .	<a href="#">13</a> , <a href="#">1006</a>
pdfcreationdate (hypersetup key) . . . . .	<a href="#">13</a> , <a href="#">2107</a>
pdfcreator (hypersetup key) . . . . .	<a href="#">2051</a>
\pdfdest . . . . .	<a href="#">11</a>
pdfdict commands:	
\pdfdict_new:n . . . . .	<a href="#">464</a> , <a href="#">619</a> , <a href="#">656</a> , <a href="#">715</a> , <a href="#">772</a>
\pdfdict_put:nnn . . . . .	<a href="#">211</a> , <a href="#">218</a> , <a href="#">465</a> , <a href="#">620</a> , <a href="#">621</a> , <a href="#">632</a> , <a href="#">635</a> , <a href="#">657</a> , <a href="#">658</a> , <a href="#">670</a> , <a href="#">677</a> , <a href="#">687</a> , <a href="#">697</a> , <a href="#">716</a> , <a href="#">717</a> , <a href="#">728</a> , <a href="#">743</a> , <a href="#">773</a> , <a href="#">774</a> , <a href="#">784</a> , <a href="#">1550</a> , <a href="#">1551</a> , <a href="#">1555</a> , <a href="#">1556</a> , <a href="#">2267</a> , <a href="#">2277</a> , <a href="#">2281</a> , <a href="#">2284</a> , <a href="#">2286</a> , <a href="#">2305</a> , <a href="#">2314</a> , <a href="#">2316</a>
\pdfdict_remove:nn . . . . .	<a href="#">208</a> , <a href="#">740</a> , <a href="#">1560</a> , <a href="#">1561</a>
\pdfdict_use:n . . . . .	<a href="#">641</a> , <a href="#">700</a> , <a href="#">747</a> , <a href="#">753</a> , <a href="#">786</a> , <a href="#">791</a> , <a href="#">2256</a>
pdfencoding (hypersetup key) . . . . .	<a href="#">1314</a>
\pdfescapestring . . . . .	<a href="#">2936</a>
pdffile commands:	
\pdffile_embed_file:nnn . . . . .	<a href="#">672</a>
PDFForm internal commands:	
\PDFForm_hypName . . . . .	<a href="#">2975</a> , <a href="#">2978</a> , <a href="#">2982</a> , <a href="#">2985</a>
pdfinfo (hypersetup key) . . . . .	<a href="#">2169</a>
pdfkeywords (hypersetup key) . . . . .	<a href="#">2051</a>
pdflang (hypersetup key) . . . . .	<a href="#">13</a> , <a href="#">2037</a>
pdflinkmargin (hypersetup key) . . . . .	<a href="#">13</a>
pdfmanagement commands:	
\pdfmanagement_add:nn . . . . .	<a href="#">1533</a>
\pdfmanagement_add:nnn . . . . .	<a href="#">500</a> , <a href="#">1084</a> , <a href="#">1085</a> , <a href="#">1107</a> , <a href="#">1409</a> , <a href="#">1434</a> , <a href="#">1450</a> , <a href="#">1454</a> , <a href="#">1474</a> , <a href="#">1482</a> , <a href="#">1511</a> , <a href="#">1566</a> , <a href="#">1576</a> , <a href="#">1600</a> , <a href="#">1614</a> , <a href="#">1626</a> , <a href="#">1631</a> , <a href="#">1656</a> , <a href="#">1664</a> , <a href="#">1679</a> , <a href="#">1705</a> , <a href="#">1730</a> , <a href="#">1762</a> , <a href="#">1779</a> , <a href="#">1825</a> , <a href="#">1850</a> , <a href="#">1864</a> , <a href="#">1884</a> , <a href="#">1911</a> , <a href="#">1940</a> , <a href="#">2046</a> , <a href="#">2065</a> , <a href="#">2083</a> ,

2094, 2118, 2132, 2150, 2239, 2258, 2430, 2447, 2454, 2456, 2458, 2464	\providecommand . . . . . 140, 141, 2366, 2369, 2596, 2932
\pdfmanagement_if_active_p: . . . 350	\ProvidesFile . . . . . 3
\pdfmanagement_remove:nn . . . . .	
. . . 1406, 1430, 1438, 1458, 1470, 1495, 1507, 1515, 1529, 1538, 1582, 1597, 1616, 1653, 1692, 1717, 1743, 1758, 1792, 1822, 1847, 1860, 1869, 1897, 1924, 1936, 1945, 2043, 2059, 2077, 2115, 2129, 2236, 2249, 2468	
\pdfmanagement_show:n . . . . . 2431	
pdfmeta commands:	
\pdfmeta_standard_verify:nnTF . . .	
. . . . . 781, 2730, 2809	
\pdfmeta_standard_verify:nTF . . .	
. . . . . 2460, 3043	
pdfmoddate (hypersetup key) . . . . 13, 2107	
pdfproducer (hypersetup key) . . . . 2051	
pdfremotestartview (hypersetup key) . . 11	
pdfstartview (hypersetup key) . . . . 11	
\pdfstringdef . . . . . 8, 9, 17, 26, 2992	
\pdfstringdefDisableCommands . . . . .	
. . . . . 323, 324, 325	
pdfsubject (hypersetup key) . . . . . 2051	
pdftitle (hypersetup key) . . . . . 2051	
pdftrapped (hypersetup key) . . . . . 2141	
pdfversion (hypersetup key) . . . . . 1314	
pdfview (hypersetup key) . . . . . 11, 1955	
pdfxform commands:	
\pdfxform_new:nnn . . . . . 2334	
\pdfxform_ref:n . . . . .	
. . . 2694, 2695, 2789, 2790, 3074, 3075	
\phantom . . . . . 2491, 2515	
\phantomsection . . . . . 3464	
plainpages (hypersetup key) . . . . . 1346	
prg commands:	
\prg_generate_conditional_-	
variant:Nnn . . . . . 369	
\prg_new_eq_conditional:NNn . . . 368	
prop commands:	
\prop_const_from_keyval:Nn . . . . .	
. . . . . 402, 410, 3525, 3540, 3555, 3570, 3585, 3600, 3616, 3631	
\prop_gput:Nnn . . . . . 11, 2352, 2359	
\prop_if_empty:NTF . . . . . 2433	
\prop_item:Nn . . . . . 2450	
\prop_map_inline:Nn . . . . .	
. . . . . 860, 924, 960, 987, 995, 1006, 1033, 1041, 1110, 1214, 1250, 1260, 1304, 1392, 2428, 2435	
\prop_new:N . . . . . 2338, 2339	
\protect 239, 254, 275, 288, 302, 374, 596, 646, 653, 707, 713, 757, 763, 806, 3397	
	\ReadBookmarks . . . . . 3386
	ref commands:
	\ref_if_exist:nn . . . . . 368
	\ref_label:nn . . . . . 364
	\ref_value:nn . . . . . 22, 385
	regex commands:
	\regex_const:Nn . . . . . 451
	\regex_extract_once:NnNTF 1805, 1836
	\relax . . . . . 596, 646, 653, 707, 713, 757, 763, 806, 2599, 2600, 2641, 2929, 2976, 2980, 3050, 3054, 3157, 3158, 3166, 3170, 3197, 3201, 3281, 3289, 3316, 3324, 3448, 3457
	\RemoveFromHook . . . . . 1414
	\RenewDocumentCommand . . . . . 155
	\RequirePackage . . . . . 6, 18, 354, 3384
	run (hypersetup key) . . . . . 9, 1392
	runborder (hypersetup key) . . . . . 13
	runborderstyle (hypersetup key) . 13, 1006
	runcolor (hypersetup key) . . . . . 896
	S
	\selectfont . . . . . 2478, 2490, 2502, 2514
	seq commands:
	\seq_const_from_clist:Nn . . . . . 400
	\seq_count:N . . . . . 1964, 1990
	\seq_get_right:NN . . . . . 1966
	\seq_item:Nn . . . . . 1807, 1838, 1960, 2008, 2010, 2020, 2025
	\seq_map_inline:Nn . . . . .
	. 437, 441, 445, 896, 1191, 1287, 2445
	\seq_new:N . . . . . 390
	\seq_put_right:Nn . . . . .
	. . . . . 1997, 1998, 1999, 2437
	\seq_set_split:Nnn . . . . . 1959
	\seq_sort:Nn . . . . . 2439
	\setlength . . . . . 3414, 3485
	\setpdflinkmargin . . . . . 8
	\settowidth . . . . . 2611
	\show . . . . . 2672
	sort commands:
	\sort_return_same: . . . . . 2443
	\sort_return_swapped: . . . . . 2442
	\space . . . . . 378, 2680, 3159, 3176, 3177, 3207, 3208, 3282, 3314, 3317
	\spacefactor . . . . . 516, 538
	\stockwidth . . . . . 3
	str commands:
	\c_colon_str . . . . . 767

<code>\c_hash_str</code> .....	637	<code>\ifundefined</code> .....	2423, 2952
<code>\c_percent_str</code> .....	638	<code>\@latex@warning</code> .....	375
<code>\str_case:nnTF</code> .....	3339	<code>\@mainaux</code> .....	2365, 2368, 2388, 2407
<code>\str_case_e:nnTF</code> .....	1960	<code>\@mkboth</code> .....	3428
<code>\str_gset:Nn</code> .....	921	<code>\@nil</code> .....	2950, 2963, 2969
<code>\str_gset_eq:NN</code> .....	487	<code>\@part</code> .....	3455, 3462
<code>\str_head:n</code> .....	2152, 2157	<code>\@pdfauthor</code> .....	21
<code>\str_if_eq:nnTF</code> .....	523, 2063, 2081, 2091, 3041	<code>\@pdfborder</code> .....	140
<code>\str_if_eq_p:nn</code> .....	735	<code>\@pdfborderstyle</code> .....	141
<code>\str_lowercase:n</code> .....	1960, 2086, 2134, 2153, 2158	<code>\@savsf</code> .....	516, 538
<code>\str_new:N</code> .....	392, 397, 398, 436	<code>\@schapter</code> .....	3425, 3426
<code>\str_set:Nn</code> .....	469	<code>\@secondoftwo</code> .....	3481
<code>\str_set_convert:Nnnn</code> .....	8	<code>\@sect</code> .....	3476, 3477
<code>\str_set_eq:NN</code> .....	489	<code>\@spart</code> .....	3468, 3469
<code>\str_tail:n</code> .....	2153, 2158	<code>\@ssect</code> .....	3411, 3412
<code>\str_uppercase:n</code> .....	2146, 2152, 2157	<code>\@tempdima</code> .....	2605, 2612, 2614, 2651, 2652, 2653, 2657, 2658, 2659, 2660
<code>\string</code> .....	2366, 2369, 2390, 2408, 2952, 2956, 2967	<code>\@tempdimb</code> .....	2611, 2612
<code>\subpdfbookmark</code> .....	3377	<code>\@typeset@protect</code> .....	3397
sys commands:		<code>\@unexpandable@protect</code> .....	239, 254, 275, 288, 302
<code>\c_sys_backend_str</code> .....	3339	<code>\c@secnumdepth</code> .....	3445, 3463, 3478
<b>T</b>			
T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:			
<code>\@BOOKMARK</code> .....	3383	<code>\calc@bm@number</code> .....	3389
<code>\@CheckBox</code> .....	2832	<code>\check@bm@number</code> .....	3388
<code>\@ChoiceMenu</code> .....	2597	<code>\Fld@additionalactions</code> .....	3039, 3109, 3143, 3179, 3216, 3263, 3303, 3332
<code>\@Form</code> .....	2421	<code>\Fld@align</code> .....	3068, 3154, 3229
<code>\@PushButton</code> .....	2722	<code>\Fld@altname</code> ..	2976, 2978, 3050, 3052
<code>\@Reset</code> .....	2799	<code>\Fld@annotflags</code> .....	3063, 3116, 3149, 3185, 3224, 3273, 3308
<code>\@Submit</code> .....	2763	<code>\Fld@annotnames</code> .....	3048, 3065, 3118, 3151, 3187, 3226, 3275, 3310
<code>\@TextField</code> .....	2562	<code>\Fld@bcolor</code> .....	3087, 3089, 3170, 3172, 3201, 3203, 3237, 3250, 3252
<code>\@bookmarksopenlevel</code> .....	1425	<code>\fld@bcolor</code> .....	3158
<code>\@bsphack</code> .....	363	<code>\Fld@bordercolor</code> .....	3083, 3085, 3127, 3136, 3138, 3157, 3166, 3168, 3197, 3199, 3235, 3246, 3248, 3281, 3289, 3291, 3316, 3324, 3326
<code>\@chapter</code> .....	3437, 3438	<code>\Fld@borderstyle</code> .....	3069, 3122, 3155, 3191, 3230, 3279, 3330
<code>\@currDisplay</code> .....	2611, 2680	<code>\Fld@borderwidth</code> ..	2658, 2659, 3069, 3122, 3155, 3191, 3230, 3279, 3330
<code>\@currValue</code> .....	2678, 2694	<code>\Fld@calculate@code</code> ..	2381, 3009, 3011
<code>\@currentHref</code> .....	3418, 3431, 3442, 3472, 3489	<code>\Fld@calculate@sortkey</code> .....	2392
<code>\@curropt</code> ..	2596, 2606, 2609, 2674, 2677	<code>\Fld@cbsymbol</code> .....	3091
<code>\@empty</code> .....	2565, 2581, 2997, 3001, 3005, 3009, 3013, 3017, 3021, 3025, 3029, 3033, 3177, 3208, 3209	<code>\Fld@charsize</code> .....	2657, 3095, 3176, 3207, 3258, 3314
<code>\@endForm</code> .....	2531	<code>\Fld@checkequals</code> .....	2609, 2677
<code>\@esphack</code> .....	365	<code>\Fld@choices</code> .....	3178
<code>\@firstofone</code> ..	2372, 2938, 2943, 2960	<code>\Fld@color</code> ..	3096, 3098, 3177, 3208, 3259
<code>\@firstoftwo</code> .....	3479	<code>\Fld@default</code> .....	2564, 2581, 2599, 2642, 2672, 2673, 2835, 3209, 3213, 3214, 3261
<code>\@for</code> .....	2607, 2675		
<code>\@gobble</code> .....	2423		
<code>\@gobbletwo</code> .....	3428		
<code>\@hyper@launch</code> .....	34, 767		
<code>\@ifnextchar</code> .....	247		

\Fld@flags	3067,	\Hy@bookmarkstype	1426
3120, 3153, 3189, 3228, 3277, 3312		\Hy@chapapp	3429, 3440
\Fld@format@code	3001, 3003	\Hy@colorlink	31
\Fld@height		\Hy@currentbookmarklevel	3370
2567, 2588, 2591, 2602, 2660, 2686,		\Hy@DisableOption	147, 154
2699, 2717, 2766, 2802, 2838, 2856		\Hy@drafttrue	1337
\Fld@keystroke@code	2997, 2999	\Hy@escapeform	
\Fld@listcount	2671, 2679, 2700, 2707		2585, 2683, 2713, 2741,
\Fld@mappingname	2980, 2982, 3054, 3056		2780, 2815, 2853, 2934, 2941, 2944
\Fld@maxlen	3264, 3266	\Hy@escapestring	2936,
\Fld@menulength	2604, 2610, 2657		2938, 2943, 2946, 2947, 2950, 2951,
\Fld@name			2960, 2963, 2999, 3003, 3007, 3011,
.. 2563, 2598, 2723, 2834, 2975, 3049			3015, 3019, 3023, 3027, 3031, 3035,
\Fld@onblur@code	3017, 3019		3091, 3142, 3205, 3261, 3262, 3299
\Fld@onclick@code	3142	\Hy@finaltrue	1342
\Fld@onenter@code	3029, 3031	\Hy@FormObjects	2427, 2867, 2929
\Fld@onexit@code	3033, 3035	\Hy@gttemp	2992, 2994
\Fld@onfocus@code	3013, 3015	\Hy@href	247
\Fld@onmousedown@code	3021, 3023	\Hy@href@nextactionraw	212
\Fld@onmouseup@code	3025, 3027	\Hy@href@page	200
\Fld@pageobjref		\Hy@hypertextnamesfalse	1352
.. 2533, 2547, 2932, 3064,		\Hy@hypertextnamestrue	1351
3117, 3150, 3186, 3225, 3274, 3309		\Hy@linkfileprefix	1354
\Fld@radiosymbol	3205	\Hy@linktoc	1377
\Fld@rotation	3079,	\Hy@localanchornamefalse	1357
3081, 3125, 3132, 3134, 3156, 3162,		\Hy@localanchornamestrue	1356
3164, 3193, 3195, 3233, 3242, 3244,		\Hy@MakeCurrentHrefAuto	
3280, 3285, 3287, 3315, 3320, 3322			3413, 3429, 3440, 3470, 3484
\Fld@submitflags	3301	\Hy@naturalnamesfalse	1361
\Fld@validate@code	3005, 3007	\Hy@naturalnamestrue	1360
\Fld@value		\Hy@next	3439, 3448, 3451
.. 2565, 2581, 2600, 2641, 2642, 3262		\Hy@numberline	143, 3371
\Fld@width	2566,	\Hy@org@chapter	3437, 3452
2580, 2587, 2591, 2601, 2640, 2651,		\Hy@OutlineName	3387
2653, 2685, 2699, 2717, 2735, 2765,		\Hy@OutlineRerunCheck	3385
2776, 2801, 2813, 2837, 2847, 2855		\Hy@pageanchorfalse	1365
\Fld@X@additionalactions		\Hy@pageanchortrue	1364
.. 2996, 3041, 3044		\Hy@pdfmajorversion	149
\Form@action	3299	\Hy@pdfminorversion	148
\G@refundefinedtrue	374	\Hy@pdfstringtrue	27, 482
\H@old@part	3455, 3466	\Hy@pdfversion	2987
\H@old@schapter	3425, 3434	\Hy@plainpagesfalse	1369
\H@old@ssect	3476, 3487, 3496	\Hy@plainpagetrue	1368
\H@old@spart	3468, 3474	\Hy@pstringdef	492, 2967
\H@old@ssect	3411, 3416	\Hy@PutCatalog	498
\href@	247	\Hy@raisedlink	3405, 3430, 3441, 3471
\Hy@abspage	510	\Hy@RestoreLastskip	537
\Hy@activeanchorfalse	555	\Hy@safe@activestrue	639, 2384
\Hy@activeanchortrue	548	\Hy@SaveLastskip	517
\Hy@AtBeginDocument	2346, 2362	\Hy@secnum@part	3458, 3460, 3463
\Hy@bookmarksnumberedfalse	1418	\Hy@SectionAnchorHref	3396, 3418, 3489
\Hy@bookmarksnumberedtrue	1419	\Hy@SectionHShift	
\Hy@bookmarksopenfalse	1422		3395, 3404, 3414, 3485
\Hy@bookmarksopentrue	1423	\Hy@StepCount	2610, 2679

\Hy@temp	2385, 2400, 2949, 2950	\HyPsd@pdfencoding@unicode	146
\Hy@unicodedefalse	2989	\HyPsd@SanitizeForOutFile	3374
\Hy@unicodetrue	145	\if@files	2364, 2387, 2406
\Hy@VerboseAnchor	518	\ifFld@combo	2625, 2655
\Hy@verbosefalse	1328	\ifFld@hidden	2580, 2640
\Hy@VerboseLinkStart	588, 603	\ifFld@multiline	2568
\Hy@VerboseLinkStop	594, 616, 649, 709, 759, 797	\ifFld@popdown	2626
\Hy@verbostrue	1327	\ifFld@radio	2622, 2645
\Hy@VersionChecked	136	\ifHy@implicit	3391
\Hy@WrapperDef	514	\ifHy@pdfescapeform	2935, 2945
\Hy@xspace@end	591, 648, 708, 758, 796	\ifHy@unicode	2988
\HyAnn@AbsPageLabel	2532, 2538, 2584, 2682, 2712, 2740, 2779, 2814, 2852	\kv@set@family@handler	160
\HyAnn@Count	2536, 2537, 2540, 2541, 2542, 2543, 2544, 2545, 2549, 2553	\ltx@empty	2341, 2342, 2381, 2531, 2532, 2533, 2564
\HyField@AddToFields	2375, 2592, 2701, 2718, 2749, 2793, 2822, 2859	\ltx@gobble	2344
\HyField@afields	2341	\ltx@gobbletwo	2345
\HyField@AfterAuxOpen	2346, 2372, 2405	\ltx@IfUndefined	3436, 3446
\HyField@AuxAddToCoFields	2345, 2357, 2369, 2390	\ltx@one	2540
\HyField@AuxAddToFields	2344, 2350, 2366, 2408	\ltx@ReturnAfterFi	2968
\HyField@cofields	2342	\ltx@zero	2537
\HyField@FlagsCheckBox	2844	\m@ne	3445, 3460
\HyField@FlagsChoice	2650	\OBJ@OCG@view	142
\HyField@FlagsPushButton	2732, 2772, 2811	\p@	2614
\HyField@FlagsRadioButton	2646	\pdf@ifdraftmode	2425
\HyField@FlagsSubmit	2773	\PDF@SetupDoc	137
\HyField@FlagsText	2579	\pdf@strcmp	2441
\HyField@PDFChoices	2710	\PDFForm@Check	2853, 2858, 3060
\HyField@SetKeys	2575, 2616, 2619, 2638, 2725, 2768, 2804, 2839	\PDFForm@List	2713, 2716, 3147
\HyPat@ObjRef	3343, 3350, 3357, 3364	\PDFForm@Name	2578, 2639, 2729, 2843, 2974
\hyper@anchor	540	\PDFForm@Push	2741, 2747, 3113
\hyper@anchorend	540, 3406, 3431, 3442, 3472	\PDFForm@Radio	2683, 2689, 3182
\hyper@anchorstart	540, 3406, 3431, 3442, 3472	\PDFForm@Reset	2815, 2821, 3306
\hyper@link	31, 31, 584	\PDFForm@Submit	2780, 2787, 3271
\hyper@linkend	30, 31, 610	\PDFForm@Text	2585, 2590, 3221
\hyper@linkfile	281, 661	\protected@edef	560
\hyper@linklaunch	34, 294, 719, 770	\strip@pt	3095, 3176, 3207, 3258, 3314
\hyper@linknamed	35, 157, 776	\toks@	3416, 3422, 3487, 3493
\hyper@linkstart	30, 31, 599	\tw@	3417, 3422, 3488, 3493
\hyper@linkurl	266, 316, 623	\XR@ext	1348
\hyper@normalise	247, 262, 276, 289, 310	\z@	2605, 3156, 3162, 3193, 3280, 3285, 3315, 3320, 3458
\HyPL@Labels	500, 510	\zref@labelbylist	2542
\HyPL@StorePageLabel	498	\zref@labelbyprops	2541
\HyPsd@LoadUnicode	144	\zref@refused	2543
\HyPsd@pdfencoding	146	\texorpdfstring	27
		text commands:	
		\text_expand:n	666
		\text_purify:n	8, 469
		\textcolor	2423, 2521, 2526
		\textsf	2521, 2526
		\the	510, 2541, 2542, 2543, 2544, 2545, 2549, 2553, 2653, 2657, 2660, 3422, 3493



