

# The hyperref-generic module

## A generic driver for hyperref

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

Version 0.95h, released 2021-07-31

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

## 2 Bookmarks / outlines

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

## 3 “Metadata”

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

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

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

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

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

## 4 Dates

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

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

## 5 PDF page size (mediabox)

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

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

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

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

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

## 6 Commands to create “external” references

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

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

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

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

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

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

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

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

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

## 6.1 Special problem: links to files

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

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

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

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

## 6.2 Splits

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

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

For this reason three new commands have been create:

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

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

## 6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.
- `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_colorrunk_bool,
\l_hyp_annot_colormenu_bool,
\l_hyp_annot_ocgcolorlink_bool,
\l_hyp_annot_ocgcolorurl_bool,
\l_hyp_annot_ocgcolorfile_bool,
\l_hyp_annot_ocgcolorrunk_bool,
\l_hyp_annot_ocgcolormenu_bool,
```

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

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

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

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

### 7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

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

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

**link margin** The driver sets a default link margin—this is identical to `pdftex` and `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{pdfstandard=A-2b}
```

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

- The `/F` key is added to links and `Print` is activated, `Hidden`, `Invisible`, `NoView` are deactivated.
- `/NeedAppearances` is suppressed
- `PushButtons`, which use the action `/S/JavaScript` are suppressed.
- `ResetButtons`, which use the action `/S/ResetForm` are suppressed.
- In widget annotations, the `/AA` dictionary is suppressed.

## 13 Destinations

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

From these instructions two (`Fit` and `FitB`) don't take an argument. All others take one (`FitH`, `FitV`, `FitBH`, `FitBV`) or more (`XYZ`, `FitR`) arguments. These arguments

are normally coordinates, **XYZ** takes also a zoom factor. The coordinates are absolute coordinates in **bp** relative to the lower left corner of the PDF.

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

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

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

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

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

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

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

- `pdfstartview` and `pdfremotestartview` will pass the optional numbers or keyword after expansion as absolute coordinates and zoom. Missing numbers will be filled up with `null`.

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

- **FitR.** If no argument (separated by spaces) follows then `pdfview` will use with 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}[2021-07-31 v0.95h %
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
```

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

```
144 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
145 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
146 \legacy_if:nT { Hy@setpdfversion }
147 {
148   \msg_warning:nn { hyp }{ pdfversion-disabled }
149 }
150 \Hy@DisableOption{pdfversion}
```

`\Acrobatmenu` should use the new internal link command

```
151 \RenewDocumentCommand \Acrobatmenu { m m }
152 {
153   \hyper@linknamed {#1} {#2}
154 }
```

`\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!

```
155 % TODO should go at some time ...
156 % \kv@set@family@handler{Hyp}
157 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
158 \cs_set_protected:Npn \hypersetup #1
159 {
160   %\kvsetkeys{Hyp} {#1}
161   \keys_set:nn { hyp / setup }{ #1 }
162 }
163 % TODO for now unknown keys should only give warnings.
164 \keys_define:nn { hyp / setup }
165 {
166   unknown .code:n =
167   {
168     \msg_warning:nnxxx { hyp } { unknown-key }
169     { hyp~/~setup }{ \l_keys_key_str } { #1 }
170   }
171 }
```

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`)

```
172
173 \bool_new:N \l__hyp_href_url_encode_bool
174 \bool_new:N \l__hyp_href_url_ismap_bool
```

```

175 \tl_new:N \l__hyp_href_url_protocol_tl
176 \tl_new:N \l__hyp_href_pdf_destination_tl
177 \tl_new:N \l__hyp_href_pdf_page_tl
178 \tl_new:N \l__hyp_href_run_parameter_tl
179 \cs_new_protected:Npn \__hyp_href_url_format {\begingroup\Url}
180
181
182 \keys_define:nn { hyp / href }
183 {
184   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
185   ,format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 },
186   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
187   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
188   ,pdfremotestartview .code:n =
189   {
190     \keys_set:nn { hyp / setup }
191     { pdfremotestartview = #1 }
192   }
193   ,page .code:n =
194   {
195     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
196     \tl_set:Nn \Hy@href@page {#1}
197   }
198   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
199   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
200   ,nextactionraw .code:n =
201   { %perhaps some safety match later, see hyperref code
202     \tl_if_empty:nTF {#1}
203     {
204       \pdfdict_remove:nn{l_hyp/annot/A}{Next}
205     }
206     {
207       \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
208       \tl_set:Nn \Hy@href@nextactionraw {/Next-#1}
209       \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
210     }
211   }
212   ,afrelationship .code:n =
213   {
214     \pdfdict_put:nnx
215     { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
216   }
217 }
218
219
220 \keys_define:nn { hyp / setup }
221 {
222   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
223   ,href / urlencode .default:n = {true}
224   ,href / urlencode .initial:n = {false}
225   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
226   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
227   ,href / format .code:n = { \cs_set_eq:NN \__hyp_href_url_format #1 }
228 }

```

```

229
230 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
231
232 \DeclareRobustCommand*{\href}[1][]{%
233   \mode_leave_vertical:
234   \hook_use:n{cmd/href/before}
235   \group_begin:
236   \keys_set:nn { hyp / href } {#1}
237   \let\protect\@unexpandable@protect
238   \bool_if:NTF \l__hyp_href_url_encode_bool
239     {
240       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
241     }
242     {
243       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
244     }
245   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
246 }
247
248 \begingroup
249 \catcode'\$=6 %
250 \catcode'\#=12 %
251 \gdef\href@\$1{\expandafter\href@split$1##\}%
252 \gdef\href@split$1#$2#$3\\$4{%
253   \hyper@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
254   \endgroup
255   \hook_use:n{cmd/href/after}
256 }%
257 \endgroup
258
259 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
260
261 \DeclareRobustCommand*{\hrefurl}[1][]{
262 {
263   \mode_leave_vertical:
264   \hook_use:n{cmd/href/before}
265   \group_begin:
266   \keys_set:nn { hyp / href } {#1}
267   \let\protect\@unexpandable@protect
268   \bool_if:NTF \l__hyp_href_url_encode_bool
269     {
270       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
271     }
272     {
273       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
274     }
275   \hyper@normalise\__hyp_href_url_aux:nn}
276
277 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
278 {
279   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
280   \group_end:
281   \hook_use:n{cmd/href/after}
282 }

```

```

283
284 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
285 \DeclareRobustCommand*{\hrefpdf}[1] []
286 {
287   \mode_leave_vertical:
288   \hook_use:n{cmd/hrefpdf/before}
289   \group_begin:
290   \keys_set:nn { hyp / href } {#1}
291   \let\protect\@unexpandable@protect
292   \hyper@normalise\__hyp_href_pdf_aux:nn
293 }
294
295 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
296 {
297   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
298   \group_end:
299   \hook_use:n{cmd/hrefpdf/after}
300 }
301
302 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
303 \DeclareRobustCommand*{\hrefrun}[1] []
304 {
305   \mode_leave_vertical:
306   \hook_use:n{cmd/hrefrun/before}
307   \group_begin:
308   \keys_set:nn { hyp / href } {#1}
309   \let\protect\@unexpandable@protect
310   \hyper@normalise\__hyp_href_run_aux:nn
311 }
312
313 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
314 {
315   \exp_args:Nnno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
316   \group_end:
317   \hook_use:n{cmd/hrefrun/after}
318 }
319
320
321 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
322
323 \DeclareRobustCommand*{\url}[1] []
324 {
325   \mode_leave_vertical:
326   \hook_use:n{cmd/url/before}
327   \group_begin:
328   \keys_set:nn { hyp / href } {#1}
329   \let\protect\@unexpandable@protect
330   \bool_if:NTF \l__hyp_href_url_encode_bool
331   {
332     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
333   }
334   {
335     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
336   }

```

```

337   \hyper@normalise\__hyp_href_url_aux:n
338 }
339
340 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
341 {
342   \exp_args:Nno
343     \hyper@linkurl{\__hyp_href_url_format {#1}}
344     {\l__hyp_href_url_protocol_tl#1}
345   \group_end:
346   \hook_use:n{cmd/url/after}
347 }
348

```

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

```

349
350 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
351   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}}%
352 % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
353 \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
354 #2}
355

```

make the new commands compatible with `\pdfstringdef`:

```

356 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
357 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
358 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
359 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

## 4 Compatibility commands

### 4.1 Metadata

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

```

360 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
361 {
362   %\tl_set:cn {@#1}{#2}
363   \AddToDocumentProperties[hyperref]{#1}{#2}
364 }
365 \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.

```

366 \hook_new:n{hyp/link/cite}
367 \color_set:nmn {hyp/color/cite}{HTML}{2E7E2A}
368 \color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
369 \keys_define:nn { hyp / setup }
370 {
371   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}}
372   ,citebordercolor

```

```

373     .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}}{#1}}
374   }
375   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
376   {
377     \keys_set:nn { hyp / setup }
378     {
379       ,linkbordercolor= hyp/color/citeborder
380       ,linkcolor      = hyp/color/cite
381     }
382   }

```

## 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

```

383   \bool_lazy_and:nnF
384   { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
385   { \msg_error:nn { hyp}{ missing-resource-management } }

```

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

```

386   \legacy_if:nT { Hy@bookmarks }
387   {
388     \AddToHook{begindocument/before}[hyperref/bookmark]{\RequirePackage{bookmark}}
389   }
390   \legacy_if:nT { Hy@draft }
391   {
392     \PassOptionsToPackage{draft}{bookmark}
393   }

```

## 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 394 %
\__hyp_ref_check:nn    395 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn    396 {
397   \@bsphack
398   \ref_label:nn{#1}{#2}
399   \@esphack
400 }
401 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```

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

```

402 \prg_new_eq_conditional:Nn \__hyp_ref_if_exist:nn \ref_if_exist:nn { p , T , F , TF }
403 \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.

```

404 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
405 {
406   \__hyp_ref_if_exist:nnF {#1}{#2}
407   {
408     \protect\G@refundefinedtrue
409     \@latex@warning
410     {
411       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
412       on~page~\thepage~\space undefined
413     }
414   }
415 }
416 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

417 \cs_new:Npn \__hyp_ref_value:nn #1 #2
418 {
419   \ref_value:nn{#1}{#2}
420 }
421 \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 422 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 423 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 424 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 425 \int_new:N \l__hyp_tmpa_int
426 \str_new:N \l__hyp_tmpa_str

```

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

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

```

\l__hyp_dest_name_tmpa_tl  TODO: document and check use!
\l__hyp_uri_tmpa_tl        427 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl   428 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str_uuu\g__hyp_text_tmpa_str 429 \tl_new:N \l__hyp_filename_tmpa_tl
430 \tl_new:N \l__hyp_para_tmpa_tl
431 \str_new:N \l__hyp_text_tmpa_str
432 \str_new:N \g__hyp_text_tmpa_str

```

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

## 7.2 Constants

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

```
433 \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`

```
434 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
435 {url,link,file,menu,run}
436 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
437 {
438   URI    = url,
439   GoTo   = link,
440   GoToR  = file,
441   Named  = menu,
442   Launch = run
443 }
444 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
445 {
446   url    = URI,
447   link   = GoTo,
448   file   = GoToR,
449   menu   = Named,
450   run    = Launch
451 }
452
```

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

## 7.3 Variables

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

```
453 \tl_new:N \g__hyp_dest_pdfstartpage_tl
454 \tl_new:N \g__hyp_dest_pdfstartview_tl
455 \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
456 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
457 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
458 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
459 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
460 \tl_new:N \l__hyp_text_enc_file_print_tl
461 \tl_new:N \l__hyp_text_enc_para_print_tl
```



```

462
463 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
464 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
465 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
466 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
467 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
468 \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.

```

469 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

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

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

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

```

470 \str_new:N \g__hyp_bordercolormodel_str

```

(End definition for `\g__hyp_bordercolormodel_str`.)

## 7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrun_bool`

`\l_hyp_annot_colormenu_bool`

```

471 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

472 {

```

```

473   \bool_new:c {l_hyp_annot_color#1_bool}

```

```

474 }

```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrun_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

```

475 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

476 {

```

```

477   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}

```

```

478 }

```

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

`\l_hyp_annot_Named_bool`

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

```

479 \seq_map_inline:Nn \c_pdfannot_link_types_seq

```

```

480 {

```

```

481   \bool_new:c {l__hyp_annot_#1_bool}

```

```

482   \bool_set_true:c {l__hyp_annot_#1_bool}

```

```

483 }

```

(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.

```
484 \box_new:N \l__hyp_dest_box
```

(End definition for `\l__hyp_dest_box`.)

## 7.6 Regex

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

```
485 \regex_const:Nn \c__hyp_dest_startview_regex
486 {
487   \A\ *
488   (?:
489     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
490     |
491     (?:Fit\b|FitB\b)
492     |
493     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
494     |
495     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
496   )
497 }
```

(End definition for `\c__hyp_dest_startview_regex`.)

## 7.7 PDF dictionaries

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

```
498 \pdfdict_new:n {l__hyp_page/Trans}
499 \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`

```
500 \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`

```

501 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
502 {
503   \str_set:Nx #2 {\text_purify:n { #1 } }
504 }

```

*(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`

```

505 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
506 {
507
508 }

```

*(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`

```

509 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
510 {
511   \pdf_string_from_unicode:nVN { #1 } #2 #2
512 }

```

*(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`

```

513 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
514 {
515   \group_begin:
516   \Hy@pdfstringtrue
517   \hook_use:n {hyp/text/pdfstring}
518   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
519   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
520   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
521   \str_gset_eq:NN \g__hyp_text_tmpa_str \l__hyp_text_tmpa_str
522   \group_end:
523   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
524 }
525 \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
526 \cs_new_protected:Npn \Hy@pstringdef #1 #2
527 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }
This is a special version for info keys:
```

```
\_hyp\_text\_pdfstring\_info:nN
528 \cs_new_protected:Npn \_hyp\_text\_pdfstring\_info:nN #1 #2
529 {
530   \_hyp\_text\_pdfstring:nonN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
531 }
```

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

## 9 Pagelabels

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

The current implementation in `hyperref/hyperref` drivers:

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

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

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

**dvipdfm:** identical to `xetex`

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

**luatex** identical to `pdftex`

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-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
532 \cs_new_protected:Npn \_hyp\_PageLabels\_gpush:
533 {
534   \pdfmanagement\_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
```

```

535 }
536
537 \def\Hy@PutCatalog #1 {}
538
539
540 \legacy_if:nT { Hy@pdfpagelabels }
541 {
542   \cs_set_protected:Npn \HyPL@StorePageLabel #1
543   {
544     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
545     \__hyp_PageLabels_gppush:
546   }
547 }

```

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

## 10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

### 10.1 Anchors / destinations

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

---

```

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

```

---

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

```

\__hyp_destination:nn

548 \Hy@WrapperDef \__hyp_destination:nn #1 #2
549 {
550   \mode_if_horizontal:T { \@savs\spacefactor }
551   \Hy@SaveLastskip      %defined in hyperref
552   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
553   \__hyp_text_pdfstring:xoN
554   { \HyperDestNameFilter{#1} }

```

```

555     { \l__hyp_text_enc_dest_tl }
556     \l__hyp_tmpa_tl
557 \str_if_eq:nnTF {#2} {fitrbox}
558 {
559     \exp_args:NV
560     \pdf_destination:nnnn \l__hyp_tmpa_tl
561     { \box_wd:N \l__hyp_dest_box }
562     { \box_ht:N \l__hyp_dest_box }
563     { \box_dp:N \l__hyp_dest_box }
564 }
565 {
566     \exp_args:NV
567     \pdf_destination:nf
568     { \l__hyp_tmpa_tl }
569     { #2 }
570 }
571 \Hy@RestoreLastskip %defined in hyperref
572 \mode_if_horizontal:T { \spacefactor\@savsf }
573 }

```

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

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

```

\hyper@anchor
\hyper@anchorstart 574 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 575 \hook_new:n{hyp/anchor}
hyp/anchor 576 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 577 {
578     \exp_args:NnV
579     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
580     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
581     \hook_use:n{hyp/anchor}
582 }
583
584 \cs_new_protected:Npn \hyper@anchorstart #1
585 {
586     \Hy@activeanchortrue
587     \exp_args:NnV
588     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
589     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
590     \hook_use:n{hyp/anchor}
591 }
592
593 \cs_new_protected:Npn \hyper@anchorend
594 {
595     \Hy@activeanchorfalse
596 }

```

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

## 10.2 GoTo Links

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

At first the internal link commands:

```

597 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
598 {
599   \mode_leave_vertical:
600   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
601   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
602   {
603     \msg_warning:nnx
604     { hyp }
605     { empty-destination-name }
606     { \c__hyp_dest_undefined_tl }
607     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
608   }
609   {
610     \__hyp_text_pdfstring:xoN
611     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
612     { \l__hyp_text_enc_dest_tl }
613     \l__hyp_dest_name_tmpa_tl
614   }
615   \exp_args:No
616   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
617 }
618
619 \cs_new_protected:Npn \__hyp_link_goto_end:
620 {
621   \pdfannot_link_goto_end:
622 }
```

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

---

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

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

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

```

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

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

---

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

---

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

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

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

```

623
624 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
625 {
626   \bool_if:NTF \l__hyp_annot_GoTo_bool
627   {
628     \Hy@VerboseLinkStart{#1}{#2}
629     \group_begin:
630     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
631     \hook_use:n {hyp/link/#1}
632     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
633     \__hyp_link_goto_end:
634     \group_end:
635     \Hy@VerboseLinkStop
636   }
637   {\let\protect\relax#3}}
638 }
639
640 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
641 {
642   \bool_if:NT \l__hyp_annot_GoTo_bool
643   {
644     \Hy@VerboseLinkStart{#1}{#2}% only for debug
645     \group_begin:
646     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
647     \hook_use:n {hyp/link/#1}
648     \__hyp_link_goto_begin:nw {#2}
649   }
650 }
651
652 \cs_new_protected:Npn \hyper@linkend
653 {
654   \bool_if:NT \l__hyp_annot_GoTo_bool
655   {
656     \__hyp_link_goto_end:
657     \group_end:
658     \Hy@VerboseLinkStop
659   }
660 }
```

### 10.3 URI links

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

```

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



```

662 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
663 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
664
665 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
666 {
667   \bool_if:NTF \l__hyp_annot_URI_bool
668   {
669     \group_begin:
670     \__hyp_text_pdfstring:xoN
671     { #2}
672     { \l__hyp_text_enc_uri_print_tl }
673     \l__hyp_uri_tmpa_tl
674     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
675     \bool_if:NT \l__hyp_href_url_ismap_bool
676     {
677       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
678     }
679     \cs_set_eq:NN \# \c_hash_str
680     \cs_set_eq:NN \% \c_percent_str
681     \Hy@safe@activestrue
682     \mode_leave_vertical:
683     \pdfannot_dict_put:nxx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
684     \pdfannot_link:nxx { URI }
685     {
686     }
687     {
688       \let\protect\relax
689       #1
690       \Hy@xspace@end
691       \Hy@VerboseLinkStop %where is the start??
692     }
693     \group_end:
694   }
695   {\let\protect\relax#1}}
696 }
697

```

## 10.4 GoToR Links files

```

698 \pdfdict_new:n {l_hyp/annot/A/GoToR}
699 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
700 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
701
702
703 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
704 {
705   \bool_if:NTF \l__hyp_annot_GoToR_bool
706   {
707     \group_begin:
708     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
709     \exp_args:Nx
710     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
711     {
712       \pdfdict_put:nxx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap

```

```

713         \exp_args:Nnox
714         \pdffile_embed_file:nnn
715         {}
716         {\l__hyp_filename_tmpa_tl }
717         {__hyp_file_tl_to_str:N \l__hyp_filename_tmpa_tl }
718     }
719     \pdfdict_put:nnx
720     {l_hyp/annot/A/GoToR}
721     {F}
722     {\pdf_object_ref:e {__hyp_file_tl_to_str:N \l__hyp_filename_tmpa_tl}}
723     \__hyp_text_pdfstring:nnN
724     { #3 }
725     { \l__hyp_text_enc_dest_print_tl }
726     \l__hyp_dest_name_tmpa_tl
727     \tl_if_blank:eTF {#3}
728     {
729         \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
730         {
731             [
732                 \int_eval:n
733                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
734                 /\l__hyp_dest_pdfremotestartview_tl
735             ]
736         }
737     }
738     {
739         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
740     }
741     \mode_leave_vertical:

```

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

```

742     \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
743     \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
744     \pdfannot_link:nnx %expansion??
745     { GoToR }
746     {
747     }
748     {
749         \let\protect\relax
750         #1\Hy@xspace@end
751         \Hy@VerboseLinkStop %where is the start??
752     }
753     \group_end:
754 }
755 {{\let\protect\relax#1}}
756 }

```

## 10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

757 \pdfdict_new:n {l_hyp/annot/A/Launch}
758 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
759 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
760

```

```

761 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
762 {
763   \bool_if:NTF \l__hyp_annot_Launch_bool
764   {
765     \group_begin:
766     \__hyp_text_pdfstring:nnN
767       { #1 }
768       { \l__hyp_text_enc_file_print_tl }
769       \l__hyp_filename_tmpa_tl
770     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
771     \__hyp_text_pdfstring:noN
772       { #3 }
773       { \l__hyp_text_enc_para_print_tl }
774       \l__hyp_para_tmpa_tl
775     \bool_if:nTF
776     {
777       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
778       ||
779       \pdf_version_compare_p:Nn > {1.9}
780     }
781     {
782       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
783     }
784     {
785       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
786       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
787     }
788     \mode_leave_vertical:
789     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
790     \pdfannot_link:nxn
791       { Launch }
792       {
793         % /A
794         % <<
795         % \pdfdict_use:n {l_hyp/annot/A/Launch}
796         % >>
797       }
798       {
799         \let\protect\relax
800         #2\Hy@xspace@end
801         \Hy@VerboseLinkStop %where is the start??
802       }
803     \group_end:
804   }
805   {\let\protect\relax#2}}
806 }

```

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

```

807 \use:x
808 { % filename, anchor text, linkname
809   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
810 }
811 {

```

```

812     \hyper@linklaunch {#1}{#2}{#3}
813 }

```

## 10.6 Named links (menu)

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

```

814 \pdfdict_new:n {l_hyp/annot/A/Named}
815 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
816 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
817
818 \cs_new_protected:Npn \hyper@linknamed #1 #2 % #1 action, #2 link text
819 {
820   \bool_if:NTF \l__hyp_annot_Named_bool
821   {
822     \group_begin:
823     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
824     {
825       \mode_leave_vertical:
826       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
827       {\pdf_name_from_unicode_e:n{#1}}
828       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
829       \pdfannot_link:nnx { Named }
830       {
831         % /A
832         % <<
833         % \pdfdict_use:n { l_hyp/annot/A/Named }
834         % >>
835       }
836       {
837         #2
838         \Hy@xspace@end
839         \Hy@VerboseLinkStop
840       }
841     }
842     {
843       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
844       #2
845     }
846     \group_end:
847   }
848   {\let\protect\relax#2}
849 }
850

```

## 11 Link decorations

### 11.1 Functions to export and select colors

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

---

`\_hyp_color_export:nnN` `\_hyp_color_export:nnN {<color>} {<export format>} <tlvar>`

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).

```

851 \cs_new_protected:Npn \_hyp_color_export:nnN #1 #2 #3
852 {
853   \tl_if_head_eq_charcode:nNTF {#1}[ %]
854   {
855     \_hyp_color_export_aux:wnnN #1 {#2} #3
856   }
857   {
858     \color_export:nnN {#1} {#2} #3
859   }
860 }
861
862 \cs_new_protected:Npn \_hyp_color_export_aux:wnnN [#1] #2 #3 #4
863 {
864   \color_export:nnnn {#1}{#2}{#3}#4
865 }
866
867 \cs_generate_variant:Nn \_hyp_color_export:nnN {xVN}

```

(End definition for `\_hyp_color_export:nnN` and `\_hyp_color_export_aux:wnnN`.)

---

`\_hyp_color_select:n` `\_hyp_color_select:nN {<color>}`

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

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

```

868 \cs_new_protected:Npn \_hyp_color_select:n #1
869 {
870   \tl_if_head_eq_charcode:nNTF {#1}[ %]
871   {
872     \_hyp_color_select_aux:wn #1
873   }
874   {
875     \color_select:n {#1}
876   }
877 }
878
879 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
880 {
881   \color_select:nn {#1}{#2}
882 }
883
884 \cs_generate_variant:Nn \_hyp_color_select:n {e}

```

(End definition for `\_hyp_color_select:n` and `\_hyp_color_select_aux:wn`.)

---

```
\_hyp_color_set:nn \_hyp_color_set:nN {\ name } {\ color }
```

---

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

```
\_hyp_color_set:nn \_hyp_color_set_aux:nwn
```

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

```
885 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
886 {
887   \tl_if_head_eq_charcode:nNTF {#2}[ %]
888   {
889     \_hyp_color_set_aux:nwn { #1 } #2
890   }
891   {
892     \color_set:nn {#1} {#2}
893   }
894 }
895
896 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
897 {
898   \color_set:nnn {#1}{#2}{#3}
899 }
900
901 \cs_generate_variant:Nn \_hyp_color_set:nn {ne}
```

(End definition for `\_hyp_color_set:nn` and `\_hyp_color_set_aux:nwn`.)

## 11.2 Textcolor of links

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

```
902 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
903 {
904   \hook_gput_code:nnn
905     {pdfannot/link/#2/begin}
906     {hyp/color}
907     {
908       \bool_if:cT { l_hyp_annot_color#1_bool }
909       {
910         \group_begin:
911         \color_select:n { hyp/color/#1}
912       }
913     }
914   \hook_gput_code:nnn
915     {pdfannot/link/#2/end}
916     {hyp/color}
917     {
918       \bool_if:cT { l_hyp_annot_color#1_bool }
919       {
920         \group_end:
921       }
922     }
923 }
```

```

922     }
923 }

```

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

```

924 \keys_define:nn { hyp / setup }
925 {
926   ,colorlinks .meta:n =
927   {
928     ,pdfborder={0-0-0}
929     ,pdfborderstyle=
930     ,colorurl  =#1
931     ,colorlink =#1
932     ,colorrun  =#1
933     ,colormenu =#1
934     ,colorfile =#1
935   }
936   ,colorlinks .default:n = {true}
937 }

```

```

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)
allcolors (setup key)
938 \seq_map_inline:Nn \c__hyp_annot_types_seq
939 {
940   \keys_define:nn { hyp / setup }
941   {
942     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
943     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
944   }
945 }
946
947 \keys_define:nn { hyp / setup }
948 {
949   ,allcolors .meta:n =
950   {
951     ,urlcolor=#1
952     ,linkcolor=#1
953     ,runcolor=#1
954     ,filecolor=#1
955     ,menucolor=#1
956   }
957   ,allcolors .value_required:n = true
958 }

```

## 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*)

```

959 \keys_define:nn { hyp / setup }
960 {
961   ,bordercolormodel .choices:nn =
962   {rgb,cmyk}

```

```

963     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
964     ,bordercolormodel .initial:n ={rgb}
965   }

966 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
967 {
968   \keys_define:nn { hyp / setup }
969   {
970     #1bordercolor .code:n =
971     {
972       \tl_if_empty:nTF { ##1 }
973       {
974         \pdfannot_dict_remove:nn
975         {link/#2}
976         { C }
977       }
978       {
979         \__hyp_color_export:xVN {##1}\g__hyp_bordercolormodel_str \l__hyp_tmpa_tl
980         \pdfannot_dict_put:nnx
981         {link/#2}
982         { C }
983         { [\l__hyp_tmpa_tl] }
984       }
985     }
986   }
987 }

988 \keys_define:nn { hyp / setup }
989 {
990   ,allbordercolors .meta:n =
991   {
992     ,linkbordercolor=#1
993     ,urlbordercolor =#1
994     ,filebordercolor=#1
995     ,menubordercolor=#1
996     ,runbordercolor =#1
997   }
998   ,allbordercolors .value_required:n = true
999 }
1000
1001

```

### 11.3.2 Borderwidth and -arc

```

1002 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1003 {
1004   \keys_define:nn { hyp / setup }
1005   {
1006     #1border .code:n =
1007     {
1008       \tl_if_empty:nTF { ##1 }
1009       {
1010         \pdfannot_dict_remove:nn
1011         {link/#2}
1012         { Border }
1013       }

```



```

1014         {
1015             \pdfannot_dict_put:nnn
1016             {link/#2}
1017             { Border }
1018             { [##1] }
1019         }
1020     }
1021 }
1022 }
1023 \keys_define:nn { hyp / setup }
1024 {
1025     ,pdfborder .code:n =
1026     {
1027         \tl_if_empty:nTF { #1 }
1028         {
1029             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1030             {
1031                 \pdfannot_dict_remove:nn
1032                 {link/##2}
1033                 { Border }
1034             }
1035         }
1036         {
1037             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1038             {
1039                 \pdfannot_dict_put:nnn
1040                 {link/##2}
1041                 { Border }
1042                 { [#1] }
1043             }
1044         }
1045     }
1046     ,pdfborder .initial:n = {0~0~1},
1047 }

```

### 11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1048 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1049 {
runborderstyle (setup key) 1050     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1051     {
menuborderstyle (setup key) 1052         #1borderstyle .code:n =
1053         {
1054             \tl_if_empty:nTF { ##1 }
1055             {
1056                 \pdfannot_dict_remove:nn
1057                 {link/#2}
1058                 { BS }
1059             }
1060             {
1061                 \pdfannot_dict_put:nnn
1062                 {link/#2}

```

```

1063         { BS }
1064         { <<##1>> }
1065     }
1066 }
1067 }
1068 }
1069 \keys_define:nn { hyp / setup }
1070 {
1071     ,pdfborderstyle .code:n =
1072     {
1073         \tl_if_empty:nTF { #1 }
1074         {
1075             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1076             {
1077                 \pdfannot_dict_remove:nn
1078                 {link/##2}
1079                 { BS }
1080             }
1081         }
1082         {
1083             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1084             {
1085                 \pdfannot_dict_put:nnn
1086                 {link/##2}
1087                 { BS }
1088                 { <<##1>> }
1089             }
1090         }
1091     }
1092     ,pdfborderstyle .initial:n = {},
1093 }

```

## 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.

```

1094 \cs_new_protected:Npn \__hyp_ocg_init:
1095 {
1096     \pdf_object_new:nn { __hyp/OCG/View } { dict }
1097     \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1098     \pdf_object_new:nn { __hyp/OCG/config } { dict }
1099     \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1100     \pdf_object_write:nx { __hyp/OCG/refarray }
1101     {
1102         \pdf_object_ref:n { __hyp/OCG/View }
1103         \c_space_tl
1104         \pdf_object_ref:n { __hyp/OCG/Print }
1105     }
1106     \pdf_object_write:nn { __hyp/OCG/View }
1107     {
1108         /Type/OCG

```

```

1109         /Name(View)
1110         /Usage
1111         <<
1112             /Print <</PrintState/OFF>>~
1113             /View <</ViewState/ON >>~
1114         >>
1115     }
1116     \pdf_object_write:nn { __hyp/OCG/Print }
1117     {
1118         /Type/OCG
1119         /Name(Print)
1120         /Usage
1121         <<
1122             /Print <</PrintState/ON>>~
1123             /View <</ViewState/OFF>>~
1124         >>
1125     }
1126     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1127     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1128     \pdf_object_write:nx { __hyp/OCG/config }
1129     {
1130         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1131         /AS[
1132             <<
1133                 /Event/View
1134                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1135                 /Category[/View]
1136             >>
1137             <<
1138                 /Event/Print
1139                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1140                 /Category[/Print]
1141             >>
1142             <<
1143                 /Event/Export
1144                 /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1145                 /Category[/Print]
1146             >>
1147         ]
1148     }
1149     \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG/Print} }
1150     \cs_gset:Npn \__hyp_ocg_init: {}
1151 }

```

(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.

```

1152 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1153 {
1154     \hook_gput_code:nnn
1155     {pdfannot/link/#2/begin}
1156     {hyp/ocg}
1157     {
1158         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }

```

```

1159         {
1160             \__hyp_ocg_init:
1161             \group_begin:
1162             \hbox_set:Nw \l__hyp_tmpa_box
1163         }
1164     }
1165     \hook_gput_code:nnn
1166     {pdfannot/link/#2/end}
1167     {hyp/ocg}
1168     {
1169         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1170         {
1171             \hbox_set_end:
1172             \mbox
1173             {
1174                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1175                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1176                 \pdf_emc:
1177                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1178                 \group_begin:
1179                 \color_select:n { hyp/color/#1 }
1180                 \box_use_drop:N \l__hyp_tmpa_box
1181                 \group_end:
1182                 \pdf_emc:
1183             }
1184             \group_end:
1185         }
1186     }
1187 }

```

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

```

1188 \bool_lazy_or:nnTF
1189 { \pdf_version_compare_p:Nn > {1.4} }
1190 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1191 {
1192     \keys_define:nn { hyp / setup }
1193     {
1194         ,_ocgcolorlinks .meta:n =
1195         {
1196             ocgcolorlink=#1,
1197             ocgcolorurl=#1,
1198             ocgcolorfile=#1,
1199             ocgcolorrun=#1,
1200             ocgcolormenu=#1
1201         }
1202         ,_ocgcolorlinks .default:n = true
1203     }
1204 }
1205 {
1206     \keys_define:nn { hyp / setup }
1207     {
1208         ,_ocgcolorlinks .code:n =
1209         {
1210             \msg_warning:nnxx

```

```

1211         { hyp }
1212         { ignore-deprecated-or-unknown-option-in-pdf-version }
1213         { ocgcolorlinks } { \pdf_version_major:\pdf_version_minor: }
1214     }
1215 }
1216 }
1217
1218 \keys_define:nn { hyp / setup }
1219 {
1220     ,ocgcolorlinks .choice:
1221     ,ocgcolorlinks / true .meta:n =
1222     {
1223         pdfborder      = {0~0~0},
1224         pdfborderstyle = {},
1225         colorlinks     = false,
1226         _ocgcolorlinks = true
1227     }
1228     ,ocgcolorlinks / false .meta:n =
1229     {
1230         _ocgcolorlinks = false
1231     }
1232     ,ocgcolorlinks .default:n = {true}
1233 }
1234
1235 \seq_map_inline:Nn \c__hyp_annot_types_seq
1236 {
1237     \bool_lazy_or:nnTF
1238     { \pdf_version_compare_p:Nn > {1.4} }
1239     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1240     {
1241         \keys_define:nn { hyp / setup }
1242         {
1243             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1244         }
1245     }
1246     {
1247         \keys_define:nn { hyp / setup }
1248         {
1249             ,ocgcolor#1 .code:n=
1250             {
1251                 \msg_warning:nxxx
1252                 { hyp }
1253                 { ignore-deprecated-or-unknown-option-in-pdf-version }
1254                 { ocgcolor#1 }
1255                 { \pdf_version_major:\pdf_version_minor: }
1256             }
1257         }
1258     }
1259 }

```

## 11.5 Highlighting

This keys set what happens if you click on a link

```

1260 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop

```

```

1261 {
1262   \keys_define:nn { hyp / setup }
1263   {
1264     ,#1highlight .choices:nn =
1265     { /I, /N, /O, /P}
1266     {
1267       \pdfannot_dict_put:nnn
1268       {link/#2}
1269       { H }
1270       { ##1 }
1271     }
1272     ,#1highlight / .code:n =
1273     {
1274       \pdfannot_dict_remove:nn
1275       {link/#2}
1276       { H }
1277     }
1278     ,#1highlight / unknown .code:n =
1279     {
1280       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1281       { pdfhighlight }
1282       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1283       { \exp_not:n {##1} }
1284     }
1285   }
1286 }
1287
1288 }
1289
1290
1291 \keys_define:nn { hyp / setup }
1292 {
1293   ,pdfhighlight .choices:nn =
1294   { /I, /N, /O, /P}
1295   {
1296     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1297     {
1298       \pdfannot_dict_put:nnn
1299       {link/####2}
1300       { H }
1301       { #1 }
1302     }
1303   }
1304   ,pdfhighlight / .code:n =
1305   {
1306     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1307     {
1308       \pdfannot_dict_remove:nn
1309       {link/##2}
1310       { H }
1311     }
1312   }
1313   ,pdfhighlight .initial:n = {/I},
1314   ,pdfhighlight / unknown .code:n =

```

```

1315     {
1316       \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1317       { pdfhighlight }
1318       { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1319       { \exp_not:n {#1} }
1320     }
1321   }

```

## 11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1322 \keys_define:nn { hyp / setup }
hideurl (setup key) 1323 {
hidefile (setup key) 1324   hidelinks .meta:n =
hiderun (setup key) 1325   {
hidemenu (setup key) 1326     ,colorlinks      = false
1327     ,ocgcolorlinks = false
1328     ,pdfborder      = { 0~0~0 }
1329     ,pdfborderstyle=
1330   }
1331 }
1332
1333 \seq_map_inline:Nn \c__hyp_annot_types_seq
1334 {
1335   \keys_define:nn { hyp / setup }
1336   {
1337     hide#1 .meta:n =
1338     {
1339       ,color#1      = false
1340       ,ocgcolor#1   = false
1341       ,#1border      = { 0~0~0 }
1342       ,#1borderstyle =
1343     }
1344   }
1345 }

```

## 11.7 color schemes and settings

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

```

colorscheme (setup key)
1346 \keys_define:nn { hyp / setup }
1347 {
1348   colorscheme .code:n =
1349   {
1350     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1351     {
1352       \keys_set:nn { hyp /setup }
1353       {
1354         ##1 = ##2
1355       }
1356     }

```

```

1357     }
1358   }
1359   \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) 1360 \keys_define:nn { hyp / setup }
pdfversion (setup key) 1361 {
1362   ,unicode .code:n = {}
1363   ,pdfencoding .code:n = {}
1364   ,pdfversion .code:n =
1365   {
1366     \msg_warning:nn { hyp }{ pdfversion-disabled }
1367   }
1368 }
1369 %

```

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

This keys are typically set only once.

```

verbose (setup key)
debug (setup key) 1370 \keys_define:nn { hyp / setup }
draft (setup key) 1371 {
final (setup key) 1372   ,verbose .choice:
1373   ,verbose / true .code:n = { \Hy@verbosetrue}
1374   ,verbose / false .code:n = { \Hy@verbosefalse}
1375   ,verbose .default:n = {true}
1376   ,debug .meta:n = {verbose=#1}
1377   ,debug .default:n = {true}
1378 }
1379 \keys_define:nn { hyp / setup }
1380 {
1381   ,draft .code:n =
1382   {
1383     \Hy@drafttrue
1384     \PassOptionsToPackage{draft}{bookmark}
1385   }
1386   ,final .code:n =
1387   {
1388     \Hy@finaltrue
1389     \PassOptionsToPackage{final}{bookmark}
1390   }
1391 }

extension (setup key)
hypertextnames (setup key) 1392 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1393 {
pageanchor (setup key)
linktoc (setup key)
linktocpage (setup key)
plainpages (setup key)
localanchorname (setup key)
linkfileprefix (setup key)

```



```

1394 ,extension .tl_set:N = \XR@ext
1395 ,extension .initial:n= pdf
1396 ,hypertextnames .choice:
1397 ,hypertextnames / true .code:n = { \Hy@hypertextnamestrue}
1398 ,hypertextnames / false .code:n = { \Hy@hypertextnamesfalse}
1399 ,hypertextnames .default:n = {true}
1400 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1401 ,localanchorname .choice:
1402 ,localanchorname / true .code:n = { \Hy@localanchornamestrue }
1403 ,localanchorname / false .code:n = { \Hy@localanchornamefalse }
1404 ,localanchorname .default:n = {true}
1405 ,naturalnames .choice:
1406 ,naturalnames / true .code:n = { \Hy@naturalnamestrue}
1407 ,naturalnames / false .code:n = { \Hy@naturalnamesfalse}
1408 ,naturalnames .default:n = {true}
1409 ,pageanchor .choice:
1410 ,pageanchor / true .code:n = { \Hy@pageanchortrue}
1411 ,pageanchor / false .code:n = { \Hy@pageanchorfalse}
1412 ,pageanchor .default:n = {true}
1413 ,plainpages .choice:
1414 ,plainpages / true .code:n = { \Hy@plainpagestrue}
1415 ,plainpages / false .code:n = { \Hy@plainpagesfalse}
1416 ,plainpages .default:n = {true}
1417 }
1418
1419 \keys_define:nn { hyp / setup }
1420 {
1421   ,linktoc .choices:nn = { none, section, all, page }
1422   {
1423     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1424   }
1425   ,linktoc / unknown .code:n =
1426   {
1427     \msg_warning:nnxxx { hyp } { unknown-choice }
1428     { linktoc }
1429     { none, section, all, page }
1430     { \exp_not:n {#1} }
1431   }
1432   ,linktocpage .choice:
1433   ,linktocpage / true .meta:n = {linktoc=page}
1434   ,linktocpage / false .meta:n = {linktoc=section}
1435   ,linktocpage .default:n = true
1436 }
1437

```

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

```

url (setup key) 1438 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1439 {
menu (setup key) 1440   \keys_define:nn { hyp / setup }
run (setup key) 1441   {
1442     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1443   }
1444 }

```

```

1445 \keys_define:nn { hyp / setup }
1446 {
1447   ,baseurl .code:n =
1448   {
1449     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1450     \tl_if_empty:NTF \l__hyp_tmpa_tl
1451     {
1452       \pdfmanagement_remove:nn {Catalog} { URI }
1453     }
1454     {
1455       \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1456     }
1457   }
1458   %only false does something ...
1459   ,bookmarks .choice:
1460   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1461   ,bookmarks / true .code:n = {}
1462   ,bookmarks .default:n = {true}
1463   ,bookmarksnumbered .choice:
1464   ,bookmarksnumbered / false .code:n = { \Hy@bookmarksnumberedfalse }
1465   ,bookmarksnumbered / true .code:n = { \Hy@bookmarksnumberedtrue }
1466   ,bookmarksnumbered .default:n = {true}
1467   ,bookmarksopen .choice:
1468   ,bookmarksopen / false .code:n = { \Hy@bookmarksopenfalse }
1469   ,bookmarksopen / true .code:n = { \Hy@bookmarksopentruetrue }
1470   ,bookmarksopen .default:n = {true}
1471   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1472   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1473   ,pdfcenterwindow .choice:
1474   ,pdfcenterwindow / false .code:n =
1475   {
1476     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1477   }
1478   ,pdfcenterwindow / true .code:n =
1479   {
1480     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1481   }
1482   ,pdfcenterwindow / .code:n =
1483   {
1484     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1485   }
1486   ,pdfcenterwindow / unknown .code:n =
1487   {
1488     \msg_warning:nxx { hyp } { no-bool }
1489     { pdfcenterwindow }
1490     { \exp_not:n {#1} }
1491   }
1492   ,pdfcenterwindow .default:n = true
1493   ,pdfdirection .choice:
1494   ,pdfdirection / L2R .code:n =
1495   {
1496     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1497   }
1498   ,pdfdirection / R2L .code:n =

```

```

1499     {
1500         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1501     }
1502 ,pdfdirection / .code:n =
1503     {
1504         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1505     }
1506 ,pdfdirection / unknown .code:n =
1507     {
1508         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1509         { pdfdirection }
1510         { L2R , R2L }
1511         { \exp_not:n {#1} }
1512     }
1513 ,pdffitdisplaydoctitle .choice:
1514 ,pdffitdisplaydoctitle / false .code:n =
1515     {
1516         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1517     }
1518 ,pdffitdisplaydoctitle / true .code:n =
1519     {
1520         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1521     }
1522 ,pdffitdisplaydoctitle .default:n = true
1523 ,pdffduplex .choices:nn =
1524 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1525     {
1526         \pdf_version_compare:NnTF > {1.6}
1527         {
1528             \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1529             { PrintDuplex } { /#1 }
1530         }
1531         {
1532             \msg_warning:nnxx
1533             {hyp}
1534             {ignore-deprecated-or-unknown-option-in-pdf-version}
1535             {pdffduplex}
1536             {\pdf_version:}
1537         }
1538     }%
1539 ,pdffduplex / .code:n =
1540     {
1541         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1542     }
1543 ,pdffduplex / unknown .code:n =
1544     {
1545         \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1546         { pdffduplex }
1547         { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1548         { \exp_not:n {#1} }
1549     }
1550 ,pdffitwindow .choice:
1551 ,pdffitwindow / false .code:n =
1552     {

```

```

1553     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1554 }
1555 ,pdffitwindow / true .code:n =
1556 {
1557     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1558 }
1559 ,pdffitwindow / .code:n =
1560 {
1561     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1562 }
1563 ,pdffitwindow .default:n = true
1564 ,pdffitwindow / unknown .code:n =
1565 {
1566     \msg_warning:nnxx { hyp } { no-bool }
1567     { pdffitwindow }
1568     { \exp_not:n {#1} }
1569 }
1570 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1571 ,pdflinkmargin .initial:n = {1pt}
1572 ,pdfmenubar .choice:
1573 ,pdfmenubar / true .code:n =
1574 {
1575     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1576 }
1577 ,pdfmenubar / false .code:n =
1578 {
1579     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1580     { HideMenubar } { true }
1581 }
1582 ,pdfmenubar / .code:n =
1583 {
1584     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1585 }
1586 ,pdfmenubar .default:n = true
1587 ,pdfmenubar / unknown .code:n =
1588 {
1589     \msg_warning:nnxx { hyp } { no-bool }
1590     { pdfmenubar }
1591     { \exp_not:n {#1} }
1592 }
1593 ,pdfnewwindow .choice:
1594 ,pdfnewwindow / true .code:n =
1595 {
1596     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1597     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1598 }
1599 ,pdfnewwindow / false .code:n =
1600 {
1601     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1602     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1603 }
1604 ,pdfnewwindow / .code:n =
1605 {
1606     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}

```

```

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

```

```

1661 ,pdfpagelayout / .code:n =
1662 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1663 ,pdfpagelayout / unknown .code:n =
1664 {
1665   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1666   { pdfpagelayout }
1667   { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1668   { \exp_not:n {#1} }
1669 }
1670 ,pdfpagemode .choices:nn =
1671 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1672 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1673 ,pdfpagemode / UseAttachments .code:n =
1674 {
1675   \pdf_version_compare:NnTF > {1.5}
1676   {
1677     \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1678   }
1679   {
1680     \msg_warning:nnxx
1681     {hyp}
1682     {ignore-deprecated-or-unknown-value-in-pdf-version}
1683     {UseAttachments}
1684     {\pdf_version:}
1685   }
1686 }
1687 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1688 ,pdfpagemode / unknown .code:n =
1689 {
1690   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1691   { pdfpagemode }
1692   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1693   { \exp_not:n {#1} }
1694 }
1695 ,pdfpagescrop .code:n =
1696 {
1697   \tl_if_empty:nTF {#1} %or blank?
1698   {
1699     \pdfmanagement_remove:nn {Pages} { CropBox }
1700   }
1701   {
1702     \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1703   }
1704 }
1705 ,pdfpicktraybypdfsize .choice:
1706 ,pdfpicktraybypdfsize / true .code:n =
1707 {
1708   \pdf_version_compare:NnTF > {1.6}
1709   {
1710     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1711     { PickTrayByPDFSize } { true }
1712   }
1713   {
1714     \msg_warning:nnxx

```

```

1715         {hyp}
1716         {ignore-deprecated-or-unknown-option-in-pdf-version}
1717         {pdfpicktraybypdfsize}
1718         {\pdf_version:}
1719     }
1720 }
1721 ,pdfpicktraybypdfsize / false .code:n =
1722 {
1723     \pdf_version_compare:NnTF > {1.6}
1724     {
1725         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1726         { PickTrayByPDFSize } { false }
1727     }
1728     {
1729         \msg_warning:nnxx
1730         {hyp}
1731         {ignore-deprecated-or-unknown-option-in-pdf-version}
1732         {pdfpicktraybypdfsize}
1733         {\pdf_version:}
1734     }
1735 }
1736 ,pdfpicktraybypdfsize / .code:n =
1737 {
1738     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1739 }
1740 ,pdfpicktraybypdfsize / unknown .code:n =
1741 {
1742     \msg_warning:nnxx { hyp } { no-bool }
1743     { picktraybypdfsize }
1744     { \exp_not:n {#1} }
1745 }
1746 ,pdfprintarea .choices:nn =
1747 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1748 {
1749     \pdf_version_compare:NnTF < {2.0}
1750     {
1751         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1752         { PrintArea } { /#1 }
1753     }
1754     {
1755         \msg_warning:nnxx
1756         {hyp}
1757         {ignore-deprecated-or-unknown-option-in-pdf-version}
1758         {pdfprintarea}
1759         {\pdf_version:}
1760     }
1761 }%
1762 ,pdfprintarea / .code:n =
1763 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1764 ,pdfprintarea / unknown .code:n =
1765 {
1766     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1767     { pdfprintarea }
1768     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1769         { \exp_not:n {#1} }
1770     }
1771 ,pdfprintclip .choices:nn =
1772 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1773 {
1774     \pdf_version_compare:NnTF < {2.0}
1775     {
1776         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1777         { PrintClip } { /#1 }
1778     }
1779     {
1780         \msg_warning:nnxx
1781         {hyp}
1782         {ignore-deprecated-or-unknown-option-in-pdf-version}
1783         {pdfprintclip}
1784         {\pdf_version:}
1785     }
1786 }%
1787 ,pdfprintclip / .code:n =
1788 {
1789     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1790 }
1791 ,pdfprintclip / unknown .code:n =
1792 {
1793     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1794     { pdfprintclip }
1795     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1796     { \exp_not:n {#1} }
1797 }
1798 ,pdfprintpagerange .code:n =
1799 {
1800     \pdf_version_compare:NnTF > {1.6}
1801     {
1802         \tl_if_empty:nTF { #1}
1803         {
1804             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1805             { PrintPageRange }
1806         }
1807         {
1808             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1809             {PrintPageRange}{[#1]}
1810         }
1811     }
1812     {
1813         \msg_warning:nnxx
1814         {hyp}
1815         {ignore-deprecated-or-unknown-option-in-pdf-version}
1816         {pdfprintpagerange}
1817         {\pdf_version:}
1818     }
1819 }
1820 ,pdfprintscaling .choices:nn =
1821 { None, AppDefault }
1822 {

```



```

1823     \pdf_version_compare:NnTF > {1.5}
1824     {
1825         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1826         { PrintScaling } { /#1 }
1827     }
1828     {
1829         \msg_warning:nnxx
1830         {hyp}
1831         {ignore-deprecated-or-unknown-option-in-pdf-version}
1832         {pdfprintscaling}
1833         {\pdf_version:}
1834     }
1835 }%
1836 ,pdfprintscaling / .code:n =
1837 {
1838     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1839 }
1840 ,pdfprintscaling / unknown .code:n =
1841 {
1842     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1843     { pdfprintarea }
1844     { None, AppDefault }
1845     { \exp_not:n {#1} }
1846 }
1847 ,pdfremotestartview .code:n =
1848 {
1849     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1850     \exp_args:NNV
1851     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_seq
1852     {
1853         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1854     }
1855     {
1856         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1857         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1858     }
1859 }
1860 ,pdfremotestartview .initial:n = {Fit}
1861 % pdfstartpage is special as it shares code with pdfstartview
1862 ,pdfstartpage .code:n =
1863 {
1864     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1865     \bool_if:nTF
1866     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl }
1867     {
1868         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1869     }
1870     {
1871         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1872         {
1873             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartpage_tl]
1874         }
1875     }
1876 }

```

```

1877 ,pdfstartpage .initial:n =1
1878 ,pdfstartview .code:n =
1879 {
1880     \tl_set:Nx \l__hyp_tmpa_tl {#1~null~null~null~}
1881     \exp_args:NNV
1882     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1883     {
1884         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1885     }
1886     {
1887         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1888         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1889     }
1890     \bool_if:nTF
1891     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1892     {
1893         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1894     }
1895     {
1896         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1897         {
1898             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1899         }
1900     }
1901 }
1902 ,pdfstartview .initial:n = Fit
1903 ,pdftoolbar .choice:
1904 ,pdftoolbar / true .code:n =
1905 {
1906     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1907 }
1908 ,pdftoolbar / false .code:n =
1909 {
1910     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1911     { HideToolbar } { true }
1912 }
1913 ,pdftoolbar / true .code:n =
1914 {
1915     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1916 }
1917 ,pdftoolbar .default:n = true
1918 ,pdftoolbar / unknown .code:n =
1919 {
1920     \msg_warning:nxxx { hyp } { no-bool }
1921     { pdftoolbar }
1922     { \exp_not:n {#1} }
1923 }
1924 % pdfview see below.
1925 ,pdfviewarea .choices:nn =
1926 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1927 {
1928     \pdf_version_compare:NnTF < {2.0}
1929     {
1930         \pdfmanagement_add:nnn {Catalog / ViewerPreferences}

```

```

1931         { ViewArea } { /#1 }
1932     }
1933     {
1934         \msg_warning:nnxx
1935         {hyp}
1936         {ignore-deprecated-or-unknown-option-in-pdf-version}
1937         {pdfviewarea}
1938         {\pdf_version:}
1939     }
1940 }%
1941 ,pdfviewarea / .code:n =
1942 {
1943     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1944 }
1945 ,pdfviewarea / unknown .code:n =
1946 {
1947     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1948     { pdfviewarea }
1949     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1950     { \exp_not:n {#1} }
1951 }
1952 ,pdfviewclip .choices:nn =
1953 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1954 {
1955     \pdf_version_compare:NnTF < {2.0}
1956     {
1957         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1958         { ViewClip } { /#1 }
1959     }
1960     {
1961         \msg_warning:nnxx
1962         {hyp}
1963         {ignore-deprecated-or-unknown-option-in-pdf-version}
1964         {pdfviewclip}
1965         {\pdf_version:}
1966     }
1967 }%
1968 ,pdfviewclip / .code:n =
1969 {
1970     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1971 }
1972 ,pdfviewclip / unknown .code:n =
1973 {
1974     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1975     { pdfviewclip }
1976     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1977     { \exp_not:n {#1} }
1978 }
1979 ,pdfwindowui .choice:
1980 ,pdfwindowui / true .code:n =
1981 {
1982     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1983 }
1984 ,pdfwindowui / false .code:n =

```

```

1985     {
1986         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1987         { HideWindowUI } { true }
1988     }
1989     ,pdfwindowui / .code:n =
1990     {
1991         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
1992     }
1993     ,pdfwindowui / unknown .code:n =
1994     {
1995         \msg_warning:nnxx { hyp } { no-bool }
1996         { pdfwindowui }
1997         { \exp_not:n {#1} }
1998     }
1999     ,pdfwindowui .default:n = true
2000 }

```

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

```

2001 \keys_define:nn { hyp / setup }
2002 {
2003     ,pdfview .code:n =
2004     {
2005         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2006         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2007         {
2008             { xyz }
2009             {
2010                 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2011                 {
2012                     \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2013                     \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2014                     {
2015                         \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2016                     }
2017                     {
2018                         \tl_set:Nx \l__hyp_dest_pdfview_tl
2019                         {
2020                             \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2021                         }
2022                     }
2023                 }
2024                 {
2025                     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2026                 }
2027             }
2028             { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2029             { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2030             { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2031             { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2032             { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2033             { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2034             { fitr }
2035             {
2036                 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}

```

```

2037 {
2038   \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2039 }
2040 {
2041   %ensure 4 values ...
2042   \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2043   \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2044   \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2045   \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2046   \hbox_set_to_wd:Nnn \l__hyp_dest_box
2047   {
2048     \fp_eval:n
2049     {
2050       round
2051       (
2052         abs
2053         (
2054           \seq_item:Nn\l__hyp_tmpa_seq{4}
2055           -
2056           (\seq_item:Nn\l__hyp_tmpa_seq{2})
2057         ),
2058         3
2059       )
2060     }bp
2061   }{}
2062   \box_set_dp:Nn \l__hyp_dest_box
2063   {
2064     \fp_eval:n
2065     {
2066       round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2067     }bp
2068   }
2069   \box_set_ht:Nn \l__hyp_dest_box
2070   {
2071     \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2072   }
2073 }
2074 }
2075 }
2076 {
2077   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2078   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2079 }
2080 }
2081 ,pdfview .initial:n = {xyz}
2082 }

```

### 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.

```

2083 \keys_define:nn { hyp / setup }

```

```

2084 {
2085     ,pdflang .code:n =
2086     {
2087         \tl_if_empty:nTF { #1 }
2088         {
2089             \pdfmanagement_remove:nn {Catalog} { Lang }
2090         }
2091         {
2092             \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2093         }
2094         \__hyp_store_metadata:nn {pdflang}{#1}
2095     }
2096 }

```

### 12.3.1 “info keys”

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

```

pdfcreator (setup key) 2097 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfsubject (setup key) 2098 {
pdfproducer (setup key) 2099     \keys_define:nn { hyp / setup }
pdfkeywords (setup key) 2100     {
2101         pdf#1 .code:n =
2102         {
2103             \tl_if_blank:nTF {##1}
2104             {
2105                 \pdfmanagement_remove:nn {Info}{#2}
2106             }
2107             {
2108                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2109                 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2110                 {
2111                     \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2112                 }
2113             }
2114             \__hyp_store_metadata:nn {pdf#1}{##1}
2115         }
2116     }
2117     \keys_define:nn { hyp / info }
2118     {
2119         #2 .code:n =
2120         {
2121             \tl_if_blank:nTF {##1}
2122             {
2123                 \pdfmanagement_remove:nn {Info}{#2}
2124             }
2125             {
2126                 \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2127                 \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2128                 {
2129                     \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2130                 }
2131             }
2132             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}

```

```

2133     }
2134     ,unknown .code:n =
2135     {
2136         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2137         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2138         {
2139             \exp_args:Nno
2140             \pdfmanagement_add:nnx {Info}
2141             { \l_keys_key_str } {\l__hyp_tmpa_str}
2142         }
2143     }
2144 }
2145 }
2146 \__hyp_setup_info_key:nn {author} {Author}
2147 \__hyp_setup_info_key:nn {title} {Title}
2148 \__hyp_setup_info_key:nn {producer} {Producer}
2149 \__hyp_setup_info_key:nn {creator} {Creator}
2150 % ignored key: addtopdfcreator
2151 \__hyp_setup_info_key:nn {subject} {Subject}
2152 \__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.

```

2153 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2154 {
2155     \keys_define:nn { hyp / setup }
2156     {
2157         pdf#1 .code:n =
2158         {
2159             \tl_if_blank:nTF {##1}
2160             {
2161                 \pdfmanagement_remove:nn {Info}{#2}
2162             }
2163             {
2164                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2165             }
2166             \__hyp_store_metadata:nn {pdf#1}{##1}
2167         }
2168     }
2169     \keys_define:nn { hyp / info }
2170     {
2171         #2 .code:n =
2172         {
2173             \tl_if_blank:nTF {##1}
2174             {
2175                 \pdfmanagement_remove:nn {Info}{#2}
2176             }
2177             {
2178                 \pdfmanagement_add:nnx {Info}{#2}{(\tl_to_str:n {##1})}
2179             }
2180             \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2181         }
2182     }
2183 }

```

```

2184
2185 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2186 \__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 ...

```

2187 \keys_define:nn { hyp / setup }
2188 {
2189   ,pdftrapped .code:n =
2190   {
2191     \exp_args:Nne
2192     \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2193   }
2194   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2195   {
2196     \pdfmanagement_add:nnx {Info}{Trapped}
2197     {/
2198       \str_uppercase:f { \str_head:n { #1 } }
2199       \str_lowercase:f { \str_tail:n { #1 } }
2200     }
2201     \__hyp_store_metadata:nx {pdftrapped}
2202     {
2203       \str_uppercase:f { \str_head:n { #1 } }
2204       \str_lowercase:f { \str_tail:n { #1 } }
2205     }
2206   }
2207   ,_pdftrapped / unknown .code:n =
2208   {
2209     \msg_warning:nnxxx { hyp } { unknown-choice }
2210     { pdftrapped }
2211     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2212     { \exp_not:n {#1} }
2213   }
2214 }

```

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

```

2215 \keys_define:nn { hyp / setup }
2216 {
2217   pdfinfo .code:n =
2218   {
2219     \keys_set:nn { hyp / info } { #1 }
2220   }
2221 }

```

Now we set some default values

```

2222 \keys_set:nn { hyp / setup} {pdfcreator = LaTeX-with-hyperref}
2223 \keys_set:nn { hyp / setup} {pdfauthor = }
2224 \keys_set:nn { hyp / setup} {pdftitle = }
2225 \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:



```

2226 \clist_map_inline:nn
2227 {
2228   ,pdfcopyright
2229   ,pdftype
2230   ,pdflicenseurl
2231   ,pdfauthortitle
2232   ,pdfcaptionwriter
2233   ,pdfmetalang
2234   ,pdfapart
2235   ,pdfaconformance
2236   ,pdfuapart
2237   ,pdfxstandard
2238   ,pdfsource
2239   ,pdfdocumentid
2240   ,pdfinstanceid
2241   ,pdfversionid
2242   ,pdfrendition
2243   ,pdfpublication
2244   ,pdfpubtype
2245   ,pdfbytes
2246   ,pdfnumpages
2247   ,pdfissn
2248   ,pdfeissn
2249   ,pdfisbn
2250   ,pdfbookedition
2251   ,pdfpublisher
2252   ,pdfvolumenum
2253   ,pdfissuenum
2254   ,pdfpagerange
2255   ,pdfdoi
2256   ,pdfurl
2257   ,pdfidentifier
2258   ,pdfsubtitle
2259   ,pdfpubstatus
2260   ,pdfcontactaddress
2261   ,pdfcontactcity
2262   ,pdfcontactregion
2263   ,pdfcontactpostcode
2264   ,pdfcontactcountry
2265   ,pdfcontactphone
2266   ,pdfcontactemail
2267   ,pdfcontacturl
2268   ,pdfdate
2269 }
2270 {
2271   \keys_define:nn { hyp / setup }
2272   {
2273     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2274   }
2275 }
2276
2277 \str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
2278 {
2279   \exp_args:Nnx

```

```

2280 \keys_set:nn {hyp / setup}
2281 {
2282   ,pdfapart      = \pdfmeta_standard_item:n {level}
2283   ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
2284 }
2285 }

```

## 12.5 Transitions

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

```

2286 \keys_define:nn { hyp / setup }
2287 {
2288   pdfpageduration .code:n =
2289   {
2290     \tl_if_blank:nTF { #1 }
2291     {
2292       \pdfmanagement_remove:nn {Page}{Dur}
2293     }
2294     {
2295       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2296     }
2297   }
2298 }

```

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

```

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

```

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

The keys allowed in the argument of `pdfpagetransition` are

```

2299 \keys_define:nn { hyp / setup }
2300 {

```

```

2301 pdfpagetransition .code:n =
2302 {
2303     \tl_if_blank:nTF {#1}
2304     {
2305         \pdfmanagement_remove:nn {Page}{Trans}
2306     }
2307     {
2308         \group_begin:
2309         \keys_set:nn { hyp / trans }{style=R,#1}
2310         \pdf_object_unnamed_write:nx { dict }
2311         {
2312             \pdfdict_use:n {l__hyp_page/Trans}
2313         }
2314         \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2315         \group_end:
2316     }
2317 }
2318 }
2319 \keys_define:nn { hyp / trans }
2320 {
2321     ,style .choices:nn =
2322     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2323     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2324     ,style / unknown .code:n =
2325     {
2326         \msg_warning:nnxxx { hyp } { unknown-choice }
2327         { trans / style }
2328         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2329         { \exp_not:n {#1} }
2330     }
2331     ,duration .code:n =
2332     {
2333         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2334     }
2335     ,direction .choices:nn =
2336     {H,V}
2337     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2338     ,direction .choices:nn =
2339     {0,90,180,270,315}
2340     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{/ #1 } }
2341     ,direction / None .code:n =
2342     { \pdfdict_put:nnn {l__hyp_page/Trans}{ DI }{/None} }
2343     ,direction / unknown .code:n =
2344     {
2345         \msg_warning:nnxxx { hyp } { unknown-choice }
2346         { trans / direction }
2347         {
2348             H~(horizontal,~only~Split,~Blinds),
2349             V~(vertical,~only~Split,~Blinds),
2350             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2351             90~(bottom~to~top,~only~Wipe),
2352             180~(right~to~left,~only~Wipe),
2353             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2354             315~(top~left~to~bottom,~only~Glitter),

```

```

2355         None~(only~Fly)
2356     }
2357     { \exp_not:n {#1} }
2358 }
2359 ,motion .choices:nn =
2360 {I,0}
2361 { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2362 ,motion / unknown .code:n =
2363 {
2364     \msg_warning:nnxxx { hyp } { unknown-choice }
2365     { trans / motion }
2366     { I~(inwards) , 0~(outwards) }
2367     { \exp_not:n {#1} }
2368 }
2369 ,scale .code:n =
2370 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2371 ,opaque .choices:nn = {true,false}
2372 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2373 ,opaque / unknown .code:n =
2374 {
2375     \msg_warning:nnxxx { hyp } { unknown-choice }
2376     { trans / B }
2377     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2378     { \exp_not:n {#1} }
2379 }
2380 % try to set unknown keys as style
2381 ,unknown .code:n =
2382 {
2383     % warning ...
2384     \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2385 }
2386 }

```

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

```

2387 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}

```

Unfinished Form field code

```

2388 \NewDocumentCommand \MakeFieldObject { m m }
2389 {
2390     \pdfxform_new:nnn { #2 }{ } { #1 }
2391 }
2392
2393
2394 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2395 \prop_new:N \g__hyp_AcroForm_Fields_prop
2396
2397 \let\HyField@afields\ltx@empty
2398 \let\HyField@cofields\ltx@empty
2399 %% UF test for old pdftex removed
2400 %\let\HyField@AuxAddToFields\ltx@gobble
2401 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2402 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2403
2404 % the value doesn't matter, but with a prop we avoid duplicates and it is
2405 % clearly faster than removing them from a sequence

```

```

2406 \def\HyField@AuxAddToFields#1
2407 {
2408   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2409 }%
2410
2411 %fields with empty key get a value too -- lets hope that
2412 %this give the expected behaviour
2413 \def\HyField@AuxAddToCoFields #1 #2
2414 {
2415   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2416 }
2417
2418 \Hy@AtBeginDocument
2419 {
2420   \if@filesw
2421     \immediate\write\@mainaux{%
2422       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2423     }%
2424     \immediate\write\@mainaux{%
2425       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2426     }%
2427   \fi
2428   \let\HyField@AfterAuxOpen\@firstofone
2429 }%
2430
2431 \def\HyField@AddToFields
2432 {
2433   \exp_args:Nx\HyField__hypAddToFields
2434   {
2435     \pdfannot_box_ref_last:
2436   }
2437   \ifx\Fld@calculate@code\ltx@empty
2438   \else
2439     \begingroup
2440       \Hy@safe@activetrue
2441       \edef\Hy@temp{%
2442         \endgroup
2443         \if@filesw
2444           \write\@mainaux
2445           {
2446             \string\HyField@AuxAddToCoFields
2447             {
2448               \Fld@calculate@sortkey
2449             }
2450             {
2451               \pdfannot_box_ref_last:
2452             }
2453           }
2454         \fi
2455       }%
2456     \Hy@temp
2457   \fi
2458 }%
2459

```

```

2460 \def\HyField__hypAddToFields#1{
2461   \HyField@AfterAuxOpen{%
2462     \if@filesw
2463       \write\@mainaux{%
2464         \string\HyField@AuxAddToFields{#1}%
2465       }%
2466     \fi
2467   }%
2468 }%
2469
2470 \ExplSyntaxOff
2471 \ExplSyntaxOn
2472
2473 \def\@Form[#1]
2474 {
2475   \kvsetkeys{Form}{#1}
2476   \pdf@ifdraftmode{}
2477   {
2478     \Hy@FormObjects
2479     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2480     {
2481       \pdfmanagement_add:nnx { Catalog / AcroForm } { Fields }{##1}
2482       %\pdfmanagement_show:n { Catalog / AcroForm }
2483     }
2484     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2485     {
2486       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2487       {
2488         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2489       }
2490       \seq_sort:Nn \l__hyp_tmpa_seq
2491       {
2492         \str_compare:nNnTF {##1} > {##2}
2493         { \sort_return_swapped: }
2494         { \sort_return_same: }
2495       }
2496       \seq_map_inline:Nn \l__hyp_tmpa_seq
2497       {
2498         \pdfmanagement_add:nnx { Catalog / AcroForm }
2499         { CO }
2500         {
2501           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2502         }
2503       }
2504     }
2505     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2506     {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2507     \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2508     {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2509     \pdfmanagement_add:nnx {Catalog /AcroForm}
2510     {DA}{(/Helv~10~Tf~0~g)}
2511     \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2512     {
2513       \legacy_if:nT { HyField@NeedAppearances }

```

```

2514         {
2515             \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2516         }
2517     }
2518     {
2519         \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2520     }
2521 }
2522 }
2523 \ExplSyntaxOff
2524 \let\@endForm\ltx@empty
2525 \let\HyAnn@AbsPageLabel\ltx@empty
2526 \let\Fld@pageobjref\ltx@empty
2527
2528 \ExplSyntaxOn
2529 \newcount\HyAnn@Count
2530 \HyAnn@Count=\ltx@zero
2531 \def\HyAnn@AbsPageLabel
2532 {
2533     \global\advance\HyAnn@Count by\ltx@one
2534     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2535     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2536     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2537     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2538     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2539 }%
2540 \def\Fld@pageobjref
2541 {
2542     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2543     {
2544         /P~\pdf_pageobject_ref:n
2545         {
2546             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2547         }
2548     }
2549 }
2550 \ExplSyntaxOff
2551 \ExplSyntaxOn
2552 %% check if the attr should be set through
2553 %% hooks.
2554 %% check if options are missing.
2555 \def\@TextField[#1]#2{% parameters, label
2556     \def\Fld@name{#2}%
2557     \let\Fld@default\ltx@empty
2558     \let\Fld@value\@empty
2559     \def\Fld@width{\DefaultWidthofText}%
2560     \def\Fld@height{%
2561         \ifFld@multiline
2562             \DefaultHeightofTextMultiline
2563         \else
2564             \DefaultHeightofText
2565         \fi
2566     }%
2567     \beginngroup

```

```

2568 \expandafter\HyField@SetKeys\expandafter{%
2569 \DefaultOptionsofText,#1%
2570 }%
2571 \PDFForm@Name
2572 \HyField@FlagsText
2573 \ifFld@hidden\def\Fld@width{1sp}\fi
2574 \ifx\Fld@value\empty\def\Fld@value{\Fld@default}\fi
2575 \LayoutTextField{#2}{%
2576 \leavevmode
2577 \HyAnn@AbsPageLabel
2578 \Hy@escapeform\PDFForm@Text
2579 \pdfannot_box:nnnn
2580 {\Fld@width}
2581 {\Fld@height}
2582 {0pt} %is this correct?
2583 {\PDFForm@Text}
2584 \MakeTextField{\Fld@width}{\Fld@height}
2585 \HyField@AddToFields
2586 }%
2587 \endgroup
2588 }
2589 \providecommand\@curropt{}
2590 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2591 \def\Fld@name{#2}
2592 \let\Fld@default\relax
2593 \let\Fld@value\relax
2594 \def\Fld@width{\DefaultWidthofChoiceMenu}
2595 \def\Fld@height{\DefaultHeightofChoiceMenu}
2596 \begingroup
2597 \Fld@menulength=0 %
2598 \@tempdima\z@
2599 \clist_map_variable:nNn { #3 } \@curropt
2600 %\@for\@curropt:=#3\do
2601 {%
2602 \expandafter\Fld@checkequals\@curropt==\%
2603 \Hy@StepCount\Fld@menulength
2604 \settowidth{\@tempdimb}{\@currDisplay}%
2605 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2606 }%
2607 \advance\@tempdima by~15\p@
2608 \begingroup
2609 \HyField@SetKeys{#1}
2610 \edef\x{\endgroup
2611 \noexpand\expandafter
2612 \noexpand\HyField@SetKeys
2613 \noexpand\expandafter{%
2614 \expandafter\noexpand\curname DefaultOptionsof%
2615 \ifFld@radio
2616 Radio%
2617 \else
2618 \ifFld@combo
2619 \ifFld@popdown
2620 PopdownBox%
2621 \else

```



```

2622         ComboBox%
2623     \fi
2624     \else
2625         ListBox%
2626     \fi
2627 \fi
2628 \endcsname
2629 }%
2630 }\x
2631 \HyField@SetKeys{#1}%
2632 \PDFForm@Name
2633 \ifFld@hidden\def\Fld@width{1sp}\fi
2634 \ifx\Fld@value\relax
2635     \let\Fld@value\Fld@default
2636 \fi
2637 \LayoutChoiceField{#2}{%
2638     \ifFld@radio
2639         \HyField@FlagsRadioButton
2640         \__hypRadio{#3}%
2641     \else
2642         \begingroup
2643         \HyField@FlagsChoice
2644         \ifdim\Fld@width<\@tempdima
2645             \ifdim\Fld@width<1cm\@tempdima1cm\fi
2646             \edef\Fld@width{\the\Fld@width}%
2647         \fi
2648         \ifFld@combo
2649         \else
2650             \@tempdima=\the\Fld@menulength\Fld@charsize
2651             \advance\Fld@width by~\Fld@borderwidth bp %
2652             \advance\Fld@width by~\Fld@borderwidth bp %
2653             \edef\Fld@height{\the\Fld@height}%
2654         \fi
2655         \__hypListbox{#3}%
2656     \endgroup
2657 \fi
2658 }%
2659 \endgroup
2660 }
2661
2662 \def\__hypRadio#1{%
2663     \Fld@listcount=0~%
2664     %\show\Fld@default
2665     \EdefEscapeName\Fld@default{\Fld@default}%
2666     \clist_map_variable:nNn { #1 } \@curropt
2667     %\@for\@curropt:=#1\do
2668     {%
2669         \expandafter\Fld@checkequals\@curropt==\%
2670         \EdefEscapeName\@currValue{\@curropt}%
2671         \Hy@StepCount\Fld@listcount
2672         \@currDisplay\space
2673         \leavevmode
2674         \HyAnn@AbsPageLabel
2675         \Hy@escapeform\PDFForm@Radio

```

```

2676 \pdfxform_if_exist:nF { __hyp_xform_Ding }
2677 {
2678   \pdfxform_new:nnn { __hyp_xform_Ding } {}
2679   {
2680     \group_begin:
2681     \fontfamily{pzd}
2682     \fontencoding{U}
2683     \fontseries{m}
2684     \fontshape{n}
2685     \selectfont
2686     \char123
2687     \group_end:
2688   }
2689 }
2690 \pdfannot_box:nnnx
2691 {\Fld@width}
2692 {\Fld@height}
2693 {0pt} %is this correct?
2694 {
2695   \PDFForm@Radio
2696   /AP
2697   <<
2698   /N
2699   <<
2700   /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2701   %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2702   >>
2703   >>
2704 }
2705 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2706 \int_compare:nNnT { \Fld@listcount } = { 1 }
2707 { \HyField@AddToFields }
2708 \c_space_tl % deliberate space between radio buttons
2709 % to do: --> should be configurable
2710 }%
2711 }
2712
2713 \newcount\Fld@listcount
2714 \def\__hypListbox#1
2715 {
2716   \HyField@PDFChoices{#1}
2717   \mode_leave_vertical:
2718   \HyAnn@AbsPageLabel
2719   \Hy@escapeform\PDFForm@List
2720   \pdf_link_user:nnn
2721     {widget} %perhaps we need more types??
2722     {\PDFForm@List}
2723     {\MakeChoiceField{\Fld@width}{\Fld@height}}
2724   \HyField@AddToFields
2725 }
2726
2727
2728 \def\@PushButton[#1]#2{% parameters, label
2729   \def\Fld@name{#2}%

```

```

2730 \group_begin:
2731 \exp_args:No\HyField@SetKeys
2732 {
2733   \DefaultOptionsofPushButton,#1
2734 }
2735 \PDFForm@Name
2736 \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2737 {
2738   \HyField@FlagsPushButton
2739   \legacy_if:nT {Fld@hidden}
2740   {
2741     \def\Fld@width{1sp}
2742   }
2743   \LayoutPushButtonField
2744   {
2745     \mode_leave_vertical:
2746     \HyAnn@AbsPageLabel
2747     \Hy@escapeform\PDFForm@Push
2748     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2749     \pdfannot_box:nnnn
2750     {\box_wd:N\l_tmpa_box}
2751     {\box_ht:N\l_tmpa_box}
2752     {\box_dp:N\l_tmpa_box} %is this correct?
2753     {\PDFForm@Push}
2754     {\box_use:N\l_tmpa_box}
2755     \HyField@AddToFields
2756   }
2757 }
2758 {
2759   \msg_error:nn { hyp }{ pdfa-no-push-button }
2760   \LayoutPushButtonField
2761   {
2762     \mode_leave_vertical:
2763     \MakeButtonField{#2}
2764   }
2765 }
2766 \group_end:
2767 }
2768
2769 \def\@Submit[#1]#2
2770 {
2771   \def\Fld@width {\DefaultWidthofSubmit}
2772   \def\Fld@height{\DefaultHeightofSubmit}
2773   \group_begin:
2774   \exp_args:No\HyField@SetKeys
2775   {
2776     \DefaultOptionsofSubmit,#1
2777   }
2778   \HyField@FlagsPushButton
2779   \HyField@FlagsSubmit
2780   \legacy_if:nT { Fld@hidden }
2781   {
2782     \def\Fld@width{1sp}
2783   }

```

```

2784 \mode_leave_vertical:
2785 \HyAnn@AbsPageLabel
2786 \Hy@escapeform\PDFForm@Submit
2787 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2788 \pdfxform_if_exist:nF
2789 { __hyp_xform_Submit }
2790 {
2791   \pdfxform_new:nnn { __hyp_xform_Submit }{ }
2792   {
2793     \fbox{\color_select:n{yellow}\textsf{Submit}}
2794   }
2795   \pdfxform_new:nnn { __hyp_xform_SubmitP }{ }
2796   {
2797     \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2798   }
2799 }
2800 \pdfannot_box:nnnn
2801 {\box_wd:N\l_tmpa_box}
2802 {\box_ht:N\l_tmpa_box}
2803 {\box_dp:N\l_tmpa_box} %is this correct?
2804 {
2805   \PDFForm@Submit
2806   /AP<<
2807     /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2808     /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2809   >>
2810 }
2811 \HyField@AddToFields
2812 \box_use:N\l_tmpa_box
2813
2814 \group_end:
2815 }
2816
2817 \def\@Reset[#1]#2
2818 {
2819   \def\Fld@width {\DefaultWidthofReset}
2820   \def\Fld@height{\DefaultHeightofReset}
2821   \group_begin:
2822     \exp_args:No\HyField@SetKeys
2823     {
2824       \DefaultOptionsofReset,#1
2825     }
2826   \mode_leave_vertical:
2827   \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2828   {
2829     \HyField@FlagsPushButton
2830     \legacy_if:nT { Fld@hidden }
2831     { \def\Fld@width{1sp} }
2832     \HyAnn@AbsPageLabel
2833     \Hy@escapeform\PDFForm@Reset
2834     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2835     \pdfannot_box:nnnn
2836     {\box_wd:N\l_tmpa_box}
2837     {\box_ht:N\l_tmpa_box}

```

```

2838         {\box_dp:N\l_tmpa_box} %is this correct?
2839         { \PDFForm@Reset }
2840         \HyField@AddToFields
2841         \box_use:N \l_tmpa_box
2842     }
2843     {
2844         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2845         \MakeButtonField{#2}
2846     }
2847 \group_end:
2848 }
2849
2850 \def\@CheckBox[#1]#2
2851 {% parameters, label
2852   \def\Fld@name{#2}
2853   \def\Fld@default{0}
2854   \group_begin:
2855     \def\Fld@width {\DefaultWidthofCheckBox}
2856     \def\Fld@height{\DefaultHeightofCheckBox}
2857     \exp_args:No\HyField@SetKeys
2858     {
2859       \DefaultOptionsofCheckBox,#1
2860     }
2861     \PDFForm@Name
2862     \HyField@FlagsCheckBox
2863     \legacy_if:nT { Fld@hidden }
2864     {
2865       \def\Fld@width{1sp}
2866     }
2867     \LayoutCheckField{#2}
2868     {
2869       \mode_leave_vertical:
2870       \HyAnn@AbsPageLabel
2871       \Hy@escapeform\PDFForm@Check
2872       \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2873       {
2874         \pdfxform_new:nnn
2875         {__hyp_xform_CheckMarkYes}{-}
2876         {
2877           \group_begin:
2878             \fontfamily{pzd}
2879             \fontencoding{U}
2880             \fontseries{m}
2881             \fontshape{n}
2882             \selectfont
2883             \char51
2884             \group_end:
2885         }
2886         \pdfxform_new:nnn
2887         {__hyp_xform_CheckMarkOff}{-}
2888         {
2889           \group_begin:
2890             \fontfamily{pzd}
2891             \fontencoding{U}

```

```

2892         \fontseries{m}
2893         \fontshape{n}
2894         \selectfont
2895         \phantom{\char51} %perhaps xetex needs some small glyph ..
2896         \group_end:
2897     }
2898 }
2899 \pdfannot_box:nnnn
2900 { \Fld@width}
2901 { \Fld@height}
2902 { Opt} %is this correct?
2903 { \PDFForm@Check}
2904 \HyField@AddToFields %check if this works with xelatex ...
2905 }
2906 \group_end:
2907 }
2908 \ExplSyntaxOff
2909
2910 %hm. Should a luatex driver use type1 fonts in fields????
2911 \ExplSyntaxOn
2912 \def\Hy@FormObjects
2913 {
2914     \pdf_object_new:nn {__hyp/Encoding/pdfdoc} {dict}
2915     \pdf_object_new:nn {__hyp/Font/ZaDb} {dict}
2916     \pdf_object_new:nn {__hyp/Font/Helv} {dict}
2917     \pdf_object_write:nx {__hyp/Encoding/pdfdoc}
2918     {
2919         /Type/Encoding
2920         /Differences[
2921             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
2922             /ring/tilde
2923             \c_space_tl
2924             39/quotesingle
2925             \c_space_tl
2926             96/grave %
2927             \iow_newline:
2928             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2929             /fraction/guilsinglleft/guilsinglright/minus/perthousand
2930             /quotedblbase/quotedblleft/quotedblright/quoteleft
2931             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2932             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2933             \iow_newline:
2934             164/currency
2935             \c_space_tl
2936             166/brokenbar
2937             \c_space_tl
2938             168/dieresis/copyright/ordfeminine
2939             \c_space_tl
2940             172/logicalnot/.notdef/registered/macron/degree/plusminus
2941             /twosuperior/threesuperior/acute/mu
2942             \c_space_tl
2943             183/periodcentered/cedilla/onesuperior/ordmasculine
2944             \c_space_tl
2945             188/onequarter/onehalf/threequarters

```

```

2946         \iow_newline:
2947         192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2948         /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2949         /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2950         /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2951         /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2952         /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2953         /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2954         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2955         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
2956         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2957     ]
2958 }
2959 \pdf_object_write:nn {__hyp/Font/ZaDb }
2960 {
2961     /Type/Font
2962     /Subtype/Type1
2963     /Name/ZaDb
2964     /BaseFont/ZapfDingbats
2965 }
2966 \pdf_object_write:nx {__hyp/Font/Helv }
2967 {
2968     /Type/Font
2969     /Subtype/Type1
2970     /Name/Helv
2971     /BaseFont/Helvetica
2972     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
2973 }
2974 \global\let\Hy@FormObjects\relax
2975 }
2976 \ExplSyntaxOff
2977 \providecommand*{\Fld@pageobjref}{\}
2978 \ifcsname pdf@escapestring\endcsname
2979 \def\Hy@escapeform#1{%
2980     \ifHy@pdfescapeform
2981     \let\Hy@escapestring\pdfescapestring
2982     \else
2983     \let\Hy@escapestring\@firstofone
2984     \fi
2985 }%
2986 \Hy@escapeform{}%
2987 \else
2988 \let\Hy@escapestring\@firstofone
2989 \def\Hy@escapeform#1{%
2990     \ifHy@pdfescapeform
2991     \def\Hy@escapestring##1{%
2992         \noexpand\Hy@escapestring{\noexpand##1}%
2993     }%
2994     \edef\Hy@temp{#1}%
2995     \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
2996     \def\Hy@escapestring##1{%
2997         \@ifundefined{Hy@esc@\string##1}{%
2998             ##1%
2999             \ThisShouldNotHappen

```

```

3000         }{%
3001             \csname Hy@esc@\string##1\endcsname
3002         }%
3003     }%
3004     \else
3005         \let\Hy@escapestring\@firstofone
3006     \fi
3007 }%
3008 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3009     \ifx\#3\%
3010     \else
3011         \expandafter
3012         \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3013         \ltx@ReturnAfterFi{%
3014             \Hy__hypescapeform#3\@nil
3015         }%
3016     \fi
3017 }%
3018 \fi
3019 \def\PDFForm@Name{%
3020     \PDFForm__hypName\Fld@name
3021     \ifx\Fld@altname\relax
3022     \else
3023         \PDFForm__hypName\Fld@altname
3024     \fi
3025     \ifx\Fld@mappingname\relax
3026     \else
3027         \PDFForm__hypName\Fld@mappingname
3028     \fi
3029 }
3030 \def\PDFForm__hypName#1{%
3031     \beginngroup
3032     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3033         \ifHy@unicode
3034             \Hy@unicodetrue
3035         \fi
3036     \fi
3037     \pdfstringdef\Hy@gtemp#1%
3038     \endgroup
3039     \let#1\Hy@gtemp
3040 }
3041 \def\Fld@X@additionalactions{%
3042     \ifx\Fld@keystroke@code\@empty
3043     \else
3044         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3045     \fi
3046     \ifx\Fld@format@code\@empty
3047     \else
3048         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3049     \fi
3050     \ifx\Fld@validate@code\@empty
3051     \else
3052         /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3053     \fi

```



```

3054 \ifx\Fld@calculate@code\@empty
3055 \else
3056 /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3057 \fi
3058 \ifx\Fld@onfocus@code\@empty
3059 \else
3060 /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3061 \fi
3062 \ifx\Fld@onblur@code\@empty
3063 \else
3064 /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3065 \fi
3066 \ifx\Fld@onmousedown@code\@empty
3067 \else
3068 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3069 \fi
3070 \ifx\Fld@onmouseup@code\@empty
3071 \else
3072 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3073 \fi
3074 \ifx\Fld@onenter@code\@empty
3075 \else
3076 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3077 \fi
3078 \ifx\Fld@onexit@code\@empty
3079 \else
3080 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3081 \fi
3082 }
3083 \ExplSyntaxOn
3084 \def\Fld@additionalactions
3085 {%
3086 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{ }
3087 {
3088 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3089 {/AA<<\Fld@X@additionalactions>>}}
3090 }
3091 }
3092 \ExplSyntaxOff
3093 \def\Fld@annotnames{%
3094 /T(\Fld@name)%
3095 \ifx\Fld@altname\relax
3096 \else
3097 /TU(\Fld@altname)%
3098 \fi
3099 \ifx\Fld@mappingname\relax
3100 \else
3101 /TM(\Fld@mappingname)%
3102 \fi
3103 }
3104 \ExplSyntaxOn
3105 \def\PDFForm@Check
3106 {
3107 /Subtype/Widget

```

```

3108 ~\Fld@annotflags
3109 ~\Fld@pageobjref
3110 ~\Fld@annotnames
3111 /FT/Btn
3112 \Fld@flags
3113 /Q~\Fld@align
3114 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3115 /AP
3116 <<
3117 /N
3118 <<
3119 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3120 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3121 >>
3122 >>
3123 /MK<<
3124 \int_compare:nNf {\Fld@rotation}={0}
3125 {
3126 /R~\Fld@rotation
3127 }
3128 \tl_if_empty:NF\Fld@bordercolor
3129 {
3130 /BC[\Fld@bordercolor]
3131 }
3132 \tl_if_empty:NF\Fld@bcolor
3133 {
3134 /BG[\Fld@bcolor]
3135 }
3136 /CA(\Hy@escapestring{\Fld@cbsymbol})%
3137 >>
3138 /DA
3139 (
3140 /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3141 \tl_if_empty:NF \Fld@color
3142 {
3143 \c_space_tl \Fld@color
3144 }
3145 )
3146 /H/P
3147 \legacy_if:nTF {\Fld@checked}
3148 {
3149 /V/Yes /AS/Yes
3150 }
3151 {
3152 /V/Off /AS/Off
3153 }
3154 \Fld@additionalactions
3155 }
3156 \ExplSyntaxOff
3157 \ExplSyntaxOn
3158 \def\PDFForm@Push
3159 {
3160 /Subtype/Widget
3161 ~\Fld@annotflags

```

```

3162 ~\Fld@pageobjref
3163 ~\Fld@annotnames
3164 /FT/Btn
3165 ~\Fld@flags
3166 /H/P
3167 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3168 \bool_if:nT
3169 {
3170     !\int_compare_p:nNn {\Fld@rotation} = {0}
3171     ||
3172     \tl_if_exist_p:N \Fld@bordercolor
3173 }
3174 {
3175     /MK
3176     <<
3177     \int_compare:nNf {\Fld@rotation} = {0}
3178     {
3179         /R~\Fld@rotation
3180     }
3181     \tl_if_exist:NT \Fld@bordercolor
3182     {
3183         /BC[\Fld@bordercolor]
3184     }
3185     >>
3186 }
3187 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3188 \Fld@additionalactions
3189 }
3190
3191 \ExplSyntaxOff
3192 \def\PDFForm@List{%
3193     /Subtype/Widget%
3194     \Fld@annotflags
3195     \Fld@pageobjref
3196     \Fld@annotnames
3197     /FT/Ch%
3198     \Fld@flags
3199     /Q \Fld@align
3200     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3201     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3202         \ifx\Fld@bordercolor\relax\else 1\fi
3203         \ifx\Fld@bcolor\relax \else 1\fi
3204         \space
3205     \else
3206         /MK<<%
3207         \ifnum\Fld@rotation=\z@
3208         \else
3209             /R \Fld@rotation
3210         \fi
3211         \ifx\Fld@bordercolor\relax
3212         \else
3213             /BC[\Fld@bordercolor]%
3214         \fi
3215         \ifx\Fld@bcolor\relax

```

```

3216         \else
3217             /BG[\Fld@bcolor]%
3218         \fi
3219     >>%
3220 \fi
3221 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3222     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3223 \Fld@choices
3224 \Fld@additionalactions
3225 }
3226 \ExplSyntaxOn
3227 \def\PDFForm@Radio
3228 {
3229     /Subtype/Widget
3230     ~\Fld@annotflags
3231     ~\Fld@pageobjref
3232     ~\Fld@annotnames
3233     /FT/Btn
3234     \Fld@flags
3235     /H/P
3236     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3237     /MK<<
3238     \ifnum\Fld@rotation=\z@
3239     \else
3240         /R~\Fld@rotation
3241     \fi
3242     \ifx\Fld@bordercolor\relax
3243     \else
3244         /BC[\Fld@bordercolor]%
3245     \fi
3246     \ifx\Fld@bcolor\relax
3247     \else
3248         /BG[\Fld@bcolor]%
3249     \fi
3250     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3251     >>
3252     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3253         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3254     \ifx\Fld@default\@empty
3255         /V/Off%
3256         /DV/Off%
3257     \else
3258         /V/\Fld@default
3259         /DV/\Fld@default
3260     \fi
3261     \Fld@additionalactions
3262 }
3263 \ExplSyntaxOff
3264 \ExplSyntaxOn
3265 % Does an appeareance dict make sense here?
3266 \def\PDFForm@Text
3267 {
3268     /Subtype/Widget
3269     ~\Fld@annotflags

```

```

3270 ~\Fld@pageobjref
3271 ~\Fld@annotnames
3272 /FT/Tx
3273 ~\Fld@flags
3274 /Q~\Fld@align
3275 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3276 \bool_if:nT
3277 {
3278     !\int_compare_p:nNn {\Fld@rotation} = {0}
3279     ||
3280     \tl_if_exist_p:N \Fld@bordercolor
3281     ||
3282     \tl_if_exist_p:N \Fld@bcolor
3283 }
3284 {
3285     /MK
3286     <<
3287     \int_compare:nNf {\Fld@rotation} = {0}
3288     {
3289         /R~\Fld@rotation
3290     }
3291     \tl_if_exist:NT \Fld@bordercolor
3292     {
3293         /BC[\Fld@bordercolor]
3294     }
3295     \tl_if_exist:NT \Fld@bcolor
3296     {
3297         /BG[\Fld@bcolor]
3298     }
3299     >>
3300 }
3301 /DA
3302 (
3303     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3304     \tl_if_empty:NF {\c_space_tl\Fld@color}
3305 )
3306 /DV(\Hy@escapestring{\Fld@default})
3307 /V(\Hy@escapestring{\Fld@value})
3308 ~\Fld@additionalactions
3309 \int_compare:nNt { \Fld@maxlen}>{0}
3310 {
3311     /MaxLen~\Fld@maxlen
3312 }
3313 }
3314 \ExplSyntaxOff
3315
3316 \def\PDFForm@Submit{%
3317     /Subtype/Widget%
3318     \Fld@annotflags
3319     \Fld@pageobjref
3320     \Fld@annotnames
3321     /FT/Btn%
3322     \Fld@flags
3323     /H/P%

```

```

3324 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3325 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3326 \ifx\Fld@bordercolor\relax\else 1\fi
3327 \space
3328 \else
3329 /MK<<%
3330 \ifnum\Fld@rotation=\z@
3331 \else
3332 /R \Fld@rotation
3333 \fi
3334 \ifx\Fld@bordercolor\relax
3335 \else
3336 /BC[\Fld@bordercolor]%
3337 \fi
3338 >>%
3339 \fi
3340 /A<<%
3341 /S/SubmitForm%
3342 /F<<%
3343 /FS/URL%
3344 /F(\Hy@escapestring{\Form@action})%
3345 >>%
3346 \Fld@submitflags
3347 >>%
3348 \Fld@additionalactions
3349 }
3350 \ExplSyntaxOn
3351 \def\PDFForm@Reset{%
3352 /Subtype/Widget%
3353 \Fld@annotflags
3354 \Fld@pageobjref
3355 \Fld@annotnames
3356 /FT/Btn%
3357 \Fld@flags
3358 /H/P%
3359 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3360 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3361 \ifx\Fld@bordercolor\relax\else 1\fi
3362 \space
3363 \else
3364 /MK<<%
3365 \ifnum\Fld@rotation=\z@
3366 \else
3367 /R~\Fld@rotation
3368 \fi
3369 \ifx\Fld@bordercolor\relax
3370 \else
3371 /BC[\Fld@bordercolor]%
3372 \fi
3373 >>%
3374 \fi
3375 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3376 /A<</S/ResetForm>>%
3377 \Fld@additionalactions

```

```

3378     }%
3379
3380
3381 %these patterns are used in hyperref checks.
3382 %it is unclear if they are really useful and if a backend support is
3383 %needed.
3384 \str_case:VnF \c_sys_backend_str
3385 {
3386   { pdfmode }
3387   {
3388     \def\HyPat@ObjRef
3389     {
3390       [0-9]*[1-9][0-9]*~0~R
3391     }
3392   }
3393   { dvipdfmx }
3394   {
3395     \def\HyPat@ObjRef
3396     {
3397       @[\~]+
3398     }
3399   }
3400   { xdvipdfmx }
3401   {
3402     \def\HyPat@ObjRef
3403     {
3404       @[\~]+
3405     }
3406   }
3407 }
3408 { %also set in hyperref sty, so probably not needed.
3409   \def\HyPat@ObjRef/{.+}
3410 }
3411
3412
3413 \ExplSyntaxOff
3414 % UF: removed Hy@writebookmark
3415 %   \Hy@currentbookmarklevel{0}
3416 %   \Hy@numberline
3417 %   \__hypwritetorep
3418 %   counter{bookmark@seq@number}
3419 % removed \HyPsd@SanitizeForOutFile, not needed
3420 % removed \currentpdfbookmark, defined by bookmark,
3421 % should use \newcommand there
3422 % removed \subpdfbookmark, defined by bookmark,
3423 % should use \newcommand there
3424 % removed \belowpdfbookmark, defined by bookmark,
3425 % should use \newcommand there
3426 % removed \pdfbookmark, defined by bookmark,
3427 % \BOOKMARK
3428 % \@BOOKMARK
3429 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3430 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3431 %% removed \ReadBookmarks / unneeded with bookmark.

```

```

3432 %% removed \Hy@OutlineName
3433 %% removed \check@bm@number
3434 %% removed \calc@bm@number
3435
3436 \ifHy@implicit
3437 \else
3438   \expandafter\endinput
3439 \fi
3440 \newlength\Hy@SectionHShift
3441 \def\Hy@SectionAnchorHref#1{%
3442   \ifx\protect\@typeset@protect
3443     \Hy__hypSectionAnchor{#1}%
3444   \fi
3445 }
3446 \DeclareRobustCommand*{\Hy__hypSectionAnchor}[1]{%
3447   \leavevmode
3448   \hbox to Opt{%
3449     \kern-\Hy@SectionHShift
3450     \Hy@raisedlink{%
3451       \hyper@anchorstart{#1}\hyper@anchorend
3452     }%
3453     \hss
3454   }%
3455 }
3456 \let\H@old@ssect\@ssect
3457 \def\@ssect#1#2#3#4#5{%
3458   \Hy@MakeCurrentHrefAuto{section*}%
3459   \setlength{\Hy@SectionHShift}{#1}%
3460   \begingroup
3461     \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3462     \toks\tw@\expandafter{%
3463       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3464       #5%
3465     }%
3466   \edef\x{\endgroup
3467     \the\toks@{\the\toks\tw@}%
3468   }\x
3469 }
3470 \let\H@old@schapter\@schapter
3471 \def\@schapter#1{%
3472   \begingroup
3473     \let\@mkboth\@gobbletwo
3474     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3475     \Hy@raisedlink{%
3476       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3477     }%
3478   \endgroup
3479   \H@old@schapter{#1}%
3480 }
3481 \ltx@ifundefined{@chapter}{-}{%
3482   \let\Hy@org@chapter\@chapter
3483   \def\@chapter{%
3484     \def\Hy@next{%
3485       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%

```



```

3486     \Hy@raisedlink{%
3487         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3488     }%
3489 }%
3490 \ifnum\c@secnumdepth>\m@ne
3491     \ltx@ifundefined{if@mainmatter}%
3492     \iftrue{\csname if@mainmatter\endcsname}%
3493     \let\Hy@next\relax
3494 \fi
3495 \fi
3496 \Hy@next
3497 \Hy@org@chapter
3498 }%
3499 }
3500 \let\H@old@part\@part
3501 \begingroup\expandafter\expandafter\expandafter\endgroup
3502 \expandafter\ifx\csname chapter\endcsname\relax
3503     \let\Hy@secnum@part\z@
3504 \else
3505     \let\Hy@secnum@part\m@ne
3506 \fi
3507 \def\@part{%
3508     \ifnum\Hy@secnum@part>\c@secnumdepth
3509         \phantomsection
3510     \fi
3511     \H@old@part
3512 }
3513 \let\H@old@spart\@spart
3514 \def\@spart#1{%
3515     \Hy@MakeCurrentHrefAuto{part*}%
3516     \Hy@raisedlink{%
3517         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3518     }%
3519     \H@old@spart{#1}%
3520 }
3521 \let\H@old@sect\@sect
3522 \def\@sect#1#2#3#4#5#6[#7]#8{%
3523     \ifnum #2>\c@secnumdepth
3524         \expandafter\@firstoftwo
3525     \else
3526         \expandafter\@secondoftwo
3527     \fi
3528     {%
3529         \Hy@MakeCurrentHrefAuto{section*}%
3530         \setlength{\Hy@SectionHShift}{#3}%
3531         \begingroup
3532             \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[\@currentHref]}%
3533             \toks\tw@\expandafter{%
3534                 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3535                 #8%
3536             }%
3537         \edef\x{\endgroup
3538             \the\toks@{\the\toks\tw@}%
3539         }\x

```

```

3540 }{%
3541 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6} [{#7}] {#8}%
3542 }%
3543 }
3544 \expandafter\def\csname Parent-4\endcsname{}
3545 \expandafter\def\csname Parent-3\endcsname{}
3546 \expandafter\def\csname Parent-2\endcsname{}
3547 \expandafter\def\csname Parent-1\endcsname{}
3548 \expandafter\def\csname Parent0\endcsname{}
3549 \expandafter\def\csname Parent1\endcsname{}
3550 \expandafter\def\csname Parent2\endcsname{}
3551 \expandafter\def\csname Parent3\endcsname{}
3552 \expandafter\def\csname Parent4\endcsname{}
3553 %%
3554 %% End of file 'hgeneric-testphase.def'.
3555 </package>

3556 <*colorscheme>
3557 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3558 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3559 % linkcolor=
3560 %,filecolor=
3561 %,urlcolor=
3562 %,menucolor=
3563 %,runcolor=
3564 %,linkbordercolor=
3565 %,filebordercolor=
3566 %,urlbordercolor=
3567 %,menubordercolor=
3568 %,runbordercolor=
3569
3570 \prop_const_from_keyval:Nn \c__hyp_colorscheme_original_prop
3571 {
3572   linkcolor      = [rgb]{1,0,0}, %red
3573   filecolor      = [rgb]{0,1,1}, %cyan
3574   urlcolor       = [rgb]{1,0,1}, %magenta
3575   menucolor      = [rgb]{1, 0, 0}, %red
3576   runcolor       = [rgb]{0,1,1}, %cyan
3577   %-----
3578   linkbordercolor = [rgb]{1, 0 ,0 },
3579   filebordercolor = [rgb]{0, .5, .5},
3580   urlbordercolor  = [rgb]{0, 1, 1},
3581   menubordercolor = [rgb]{1, 0, 0},
3582   runbordercolor  = [rgb]{0, .7, .7}
3583 }
3584
3585 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3586 {
3587   linkcolor      = [rgb]{0,0.2,0.6},
3588   filecolor      = [rgb]{0.8,0,0.8},
3589   urlcolor       = [rgb]{0.8,0,0.8},
3590   menucolor      = [rgb]{0,0.2,0.6},
3591   runcolor       = [rgb]{0.8,0,0.8},
3592   %----- %-----
3593   linkbordercolor = [rgb]{0,0.2,0.6},

```

```

3594     filebordercolor = [rgb]{0.8,0,0.8},
3595     urlbordercolor  = [rgb]{0.8,0,0.8},
3596     menubordercolor = [rgb]{0,0.2,0.6},
3597     runbordercolor  = [rgb]{0.8,0,0.8}
3598 }
3599
3600 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3601 { %two colors: intern/extern
3602     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3603     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3604     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3605     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3606     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3607     %----- %-----
3608     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3609     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3610     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3611     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3612     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3613 }
3614
3615 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3616 { %all darkgray
3617     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3618     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3619     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3620     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3621     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3622     %----- %-----
3623     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3624     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3625     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3626     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3627     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3628 }
3629
3630 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3631 { %dvipsnam.def
3632     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3633     filecolor      = [rgb]{1, 0, 0}, %Red
3634     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3635     menucolor      = [rgb]{1, 0, 0}, %Red
3636     runcolor       = [rgb]{1, 0, 0}, %Red
3637     %----- %-----
3638     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3639     filebordercolor = [rgb]{1, 0, 0}, %Red
3640     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3641     menubordercolor = [rgb]{1, 0, 0}, %Red
3642     runbordercolor  = [rgb]{1, 0, 0} %Red
3643 }
3644
3645 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3646 { %dvipsnam.def
3647     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed

```

```

3648     filecolor      = [rgb]{0, 1, 0},      %Green
3649     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3650     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3651     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3652 %----- %-----
3653     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3654     filebordercolor = [rgb]{0, 1, 0},      %Green
3655     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3656     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3657     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3658 }
3659
3660
3661 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3662 {
3663     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3664     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3665     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3666     menucolor      = [rgb]{0.44706, 0.45882, 0},
3667     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3668 %----- %-----
3669     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3670     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3671     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3672     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3673     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3674 }
3675
3676 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3677 {
3678     linkcolor      = [rgb]{0,0,0},
3679     filecolor      = [rgb]{0,0,0},
3680     urlcolor       = [rgb]{0,0,0},
3681     menucolor      = [rgb]{0,0,0},
3682     runcolor       = [rgb]{0,0,0},
3683 %----- %-----
3684     linkbordercolor = [rgb]{0,0,0},
3685     filebordercolor = [rgb]{0,0,0},
3686     urlbordercolor  = [rgb]{0,0,0},
3687     menubordercolor = [rgb]{0,0,0},
3688     runbordercolor  = [rgb]{0,0,0}
3689 }
3690 </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	
\#	250, 679
\\$	249
\%	680
\.	489, 493, 495
@curropt commands:	
\@curropt:	2600, 2667
\\	17, 18, 19, 26, 30, 40, 60, 67, 74, 81, 88, 103, 108, 109, 117, 118, 127, 128, 129, 130, 131, 138, 251, 252, 809, 2602, 2669, 3009
\_	487, 489, 493, 495
A	
\A	487
\Acrobatmenu	17, 151
\addcontentsline	12
\AddToDocumentProperties	363
\AddToHook	388
\advance	2533, 2607, 2651, 2652
allcolors (hypersetup key)	<u>938</u>
\author	<u>2</u>
B	
\b	491
\begingroup	179, 248, 350, 2439, 2567, 2596, 2608, 2642, 3031, 3460, 3472, 3501, 3531
\belowpdfbookmark	3424
\bgroup	245, 350
\BOOKMARK	3427
bookmarkstype (hypersetup key)	12, <u>959</u>
bool commands:	
\bool_if:N	238, 268, 330, 626, 642, 654, 667, 675, 705, 763, 820, 908, 918, 1158, 1169
\bool_if:nTF	775, 1865, 1890, 3168, 3276
\bool_lazy_and:nnTF	383
\bool_lazy_or:nnTF	1188, 1237
\bool_new:N	173, 174, 473, 477, 481
\bool_set_true:N	482
bordercolormodel (hypersetup key)	12, <u>959</u>
box commands:	
\box_dp:N	563, 2752, 2803, 2838
\box_ht:N	562, 2751, 2802, 2837
\box_new:N	422, 484
\box_set_dp:Nn	2062
\box_set_ht:Nn	2069
\box_use:N	1175, 2754, 2812, 2841
\box_use_drop:N	1180
\box_wd:N	561, 2750, 2801, 2836
\l_tmpa_box	2748, 2750, 2751, 2752, 2754, 2787, 2801, 2802, 2803, 2812, 2834, 2836, 2837, 2838, 2841
C	
\catcode	249, 250
\char	2686, 2883, 2895
\chardef	136
\cite	31
clist commands:	
\clist_map_function:nN	110, 119
\clist_map_inline:nn	2226
\clist_map_variable:nNn	2599, 2666
color commands:	
\color_export:nnN	858
\color_export:nnnN	864
\color_select:n	875, 911, 1179, 2793, 2797
\color_select:nn	881
\color_set:nn	2, 368, 892
\color_set:nnn	2, 367, 898
color names:	
hyp/annot/file	<u>470</u>
hyp/annot/link	<u>470</u>
hyp/annot/menu	<u>470</u>
hyp/annot/run	<u>470</u>
hyp/annot/url	<u>470</u>
colorfile (hypersetup key)	<u>938</u>
colorlink (hypersetup key)	<u>938</u>
colorlinks (hypersetup key)	<u>924</u>
colormenu (hypersetup key)	<u>938</u>
colorrund (hypersetup key)	<u>938</u>
colorscheme (hypersetup key)	<u>1</u> , <u>1346</u>
colorurl (hypersetup key)	<u>938</u>
cs commands:	
\cs_generate_variant:Nn	134, 135, 365, 401, 416, 421, 525, 867, 884, 901
\cs_gset:Npn	1150
\cs_if_exist_p:N	384
\cs_new:Npn	103, 417
\cs_new_protected:Npn	179, 277, 295, 313, 340, 360, 395, 404, 501, 505, 509, 513, 526, 528, 532, 576, 584, 593, 597, 619, 624, 640,

652, 665, 703, 761, 818, 851, 862, 868, 879, 885, 896, 1094, 2097, 2153	
<code>\cs_set_eq:NN</code> . . . . .	
139, 144, 145, 185, 227, 679, 680, 1423	
<code>\cs_set_protected:Npn</code> . . . . .	
. . . . . 137, 158, 542, 809	
<code>\csname</code> . . . . . 2614, 3001,	
3012, 3492, 3502, 3544, 3545, 3546,	
3547, 3548, 3549, 3550, 3551, 3552	
<code>\currentpdfbookmark</code> . . . . . 3420	
<b>D</b>	
<code>\d</code> . . . . . 489, 493, 495	
debug (hypersetup key) . . . . . <u>1370</u>	
<code>\DeclareDocumentMetadata</code> . . . . .	
. . . . . 3, 10, 13, 14, 17, 19, 27, 33	
<code>\DeclareRobustCommand</code> . . . . .	
. . . 232, 261, 285, 303, 323, 353, 3446	
<code>\def</code> . . . . . 143, 350, 351, 353,	
537, 2402, 2406, 2413, 2431, 2460,	
2473, 2531, 2540, 2555, 2556, 2559,	
2560, 2573, 2574, 2590, 2591, 2594,	
2595, 2633, 2662, 2714, 2728, 2729,	
2741, 2769, 2771, 2772, 2782, 2817,	
2819, 2820, 2831, 2850, 2852, 2853,	
2855, 2856, 2865, 2912, 2979, 2989,	
2991, 2996, 3008, 3019, 3030, 3041,	
3084, 3093, 3105, 3158, 3192, 3227,	
3266, 3316, 3351, 3388, 3395, 3402,	
3409, 3441, 3457, 3471, 3483, 3484,	
3507, 3514, 3522, 3544, 3545, 3546,	
3547, 3548, 3549, 3550, 3551, 3552	
<code>\DefaultHeightofCheckBox</code> . . . . . 2856	
<code>\DefaultHeightofChoiceMenu</code> . . . . . 2595	
<code>\DefaultHeightofReset</code> . . . . . 2820	
<code>\DefaultHeightofSubmit</code> . . . . . 2772	
<code>\DefaultHeightofText</code> . . . . . 2564	
<code>\DefaultHeightofTextMultiline</code> . . . 2562	
<code>\DefaultOptionsofCheckBox</code> . . . . . 2859	
<code>\DefaultOptionsofPushButton</code> . . . . . 2733	
<code>\DefaultOptionsofReset</code> . . . . . 2824	
<code>\DefaultOptionsofSubmit</code> . . . . . 2776	
<code>\DefaultOptionsofText</code> . . . . . 2569	
<code>\DefaultWidthofCheckBox</code> . . . . . 2855	
<code>\DefaultWidthofChoiceMenu</code> . . . . . 2594	
<code>\DefaultWidthofReset</code> . . . . . 2819	
<code>\DefaultWidthofSubmit</code> . . . . . 2771	
<code>\DefaultWidthofText</code> . . . . . 2559	
destlabel (hypersetup key) . . . . . <u>12</u>	
dim commands:	
<code>\dim_to_decimal_in_bp:n</code> . . . . . <u>11</u> , <u>139</u>	
<code>\do</code> . . . . . 2600, 2667	
<code>\documentclass</code> . . . . . 20	
draft (hypersetup key) . . . . . <u>1370</u>	
<b>E</b>	
<code>\edef</code> 2441, 2610, 2646, 2653, 2994, 3466, 3537	
<code>\EdefEscapeName</code> . . . . . 2665, 2670	
<code>\egroup</code> . . . . . 353	
<code>\else</code> . . . . . 2438, 2563, 2617,	
2621, 2624, 2641, 2649, 2982, 2987,	
3004, 3010, 3022, 3026, 3043, 3047,	
3051, 3055, 3059, 3063, 3067, 3071,	
3075, 3079, 3096, 3100, 3201, 3202,	
3203, 3205, 3208, 3212, 3216, 3222,	
3239, 3243, 3247, 3253, 3257, 3325,	
3326, 3328, 3331, 3335, 3360, 3361,	
3363, 3366, 3370, 3437, 3504, 3525	
<code>\endcsname</code> . . . . . 2628, 2978, 3001,	
3012, 3492, 3502, 3544, 3545, 3546,	
3547, 3548, 3549, 3550, 3551, 3552	
<code>\endgroup</code> . . . . . 254, 257,	
351, 353, 2442, 2587, 2610, 2656,	
2659, 3038, 3466, 3478, 3501, 3537	
<code>\endinput</code> . . . . . 3438	
exp commands:	
<code>\exp_args:Ne</code> . . . . . 3086	
<code>\exp_args:Nne</code> . . . . . 2191	
<code>\exp_args:Nnno</code> . . . . . 297, 315	
<code>\exp_args:Nno</code> . . . . . 279, 342, 2139	
<code>\exp_args:Nnox</code> . . . . . 713	
<code>\exp_args:NNV</code> . . . . . 1850, 1881	
<code>\exp_args:NnV</code> . . . . . 578, 587	
<code>\exp_args:Nnx</code> . . . . . 2279, 2384	
<code>\exp_args:No</code> . . . . .	
. . . . 611, 615, 2731, 2774, 2822, 2857	
<code>\exp_args:NV</code> . . . . . 559, 566	
<code>\exp_args:Nx</code> . . . . 709, 2132, 2180, 2433	
<code>\exp_not:N</code> . . . . . 809	
<code>\exp_not:n</code> . . . . .	
1285, 1319, 1430, 1490, 1511, 1548,	
1568, 1591, 1635, 1668, 1693, 1744,	
1769, 1796, 1845, 1922, 1950, 1977,	
1997, 2212, 2329, 2357, 2367, 2378	
<code>\expandafter</code> . . . . . 251, 2568, 2602,	
2611, 2613, 2614, 2669, 2995, 3011,	
3438, 3462, 3463, 3501, 3502, 3524,	
3526, 3533, 3534, 3544, 3545, 3546,	
3547, 3548, 3549, 3550, 3551, 3552	
<code>\ExplSyntaxOff</code> . . . . .	
. . . 9, 2470, 2523, 2550, 2908, 2976,	
3092, 3156, 3191, 3263, 3314, 3413	
<code>\ExplSyntaxOn</code> . . . . .	
. . . 7, 10, 2471, 2528, 2551, 2911,	
3083, 3104, 3157, 3226, 3264, 3350	
extension (hypersetup key) . . . . . <u>12</u> , <u>1392</u>	
<b>F</b>	
<code>\fbox</code> . . . . . 2705, 2793, 2797	

<code>\fi</code> .....	2427, 2454, 2457, 2466, 2565, 2573, 2574, 2605, 2623, 2626, 2627, 2633, 2636, 2645, 2647, 2654, 2657, 2984, 3006, 3016, 3018, 3024, 3028, 3035, 3036, 3045, 3049, 3053, 3057, 3061, 3065, 3069, 3073, 3077, 3081, 3098, 3102, 3201, 3202, 3203, 3210, 3214, 3218, 3220, 3222, 3241, 3245, 3249, 3253, 3260, 3325, 3326, 3333, 3337, 3339, 3360, 3361, 3368, 3372, 3374, 3439, 3444, 3494, 3495, 3506, 3510, 3527		
file (hypersetup key) .....	9, <a href="#">1438</a>		
file commands:			
<code>\file_input:n</code> .....	8		
fileborderstyle (hypersetup key) .....	<a href="#">13</a> , <a href="#">1048</a>		
filecolor (hypersetup key) .....	<a href="#">938</a>		
final (hypersetup key) .....	<a href="#">1370</a>		
<code>\fontencoding</code> .....	2682, 2879, 2891		
<code>\fontfamily</code> .....	2681, 2878, 2890		
<code>\fontseries</code> .....	2683, 2880, 2892		
<code>\fontshape</code> .....	2684, 2881, 2893		
fp commands:			
<code>\fp_eval:n</code> .....	2020, 2048, 2064		
<b>G</b>			
<code>\gdef</code> .....	251, 252		
<code>\GetDocumentProperties</code> .....	2		
<code>\global</code> .....	2533, 2974		
group commands:			
<code>\group_begin:</code> ..	235, 265, 289, 307, 327, 515, 629, 645, 669, 707, 765, 822, 910, 1161, 1178, 2308, 2680, 2730, 2773, 2821, 2854, 2877, 2889		
<code>\group_end:</code> .....	280, 298, 316, 345, 522, 634, 657, 693, 753, 803, 846, 920, 1181, 1184, 2315, 2687, 2766, 2814, 2847, 2884, 2896, 2906		
<b>H</b>			
<code>\hbox</code> .....	350, 3448		
hbox commands:			
<code>\hbox_overlap_right:n</code> .....	1175		
<code>\hbox_set:Nn</code> .....	2748, 2787, 2834		
<code>\hbox_set:Nw</code> .....	1162		
<code>\hbox_set_end:</code> .....	1171		
<code>\hbox_set_to_wd:Nnn</code> .....	2046		
hidefile (hypersetup key) .....	<a href="#">1322</a>		
hidelink (hypersetup key) .....	<a href="#">1322</a>		
hidelinks (hypersetup key) .....	<a href="#">1322</a>		
hidemenu (hypersetup key) .....	<a href="#">1322</a>		
hiderun (hypersetup key) .....	<a href="#">1322</a>		
hideurl (hypersetup key) .....	<a href="#">1322</a>		
hook commands:			
<code>\hook_gput_code:nnn</code> .....	375, 904, 914, 1154, 1165		
<code>\hook_new:n</code> .....	366, 500, 575		
<code>\hook_new_pair:nn</code> .....	230, 259, 284, 302, 321		
<code>\hook_use:n</code> .....	234, 255, 264, 281, 288, 299, 306, 317, 326, 346, 517, 581, 590, 631, 647		
<code>\href</code> .....	3–5, 17, 232		
<code>\hreflaunch</code> .....	5		
<code>\hrefpdf</code> .....	4, 5, 17, 285, 358		
<code>\hrefrun</code> .....	4, 303, 359		
<code>\hrefurl</code> .....	4, 5, 17, 261, 357		
<code>\hss</code> .....	3453		
Hy internal commands:			
<code>\Hy_hyescapeform</code> ..	2995, 3008, 3014		
<code>\Hy_hypSectionAnchor</code> ...	3443, 3446		
HyField internal commands:			
<code>\HyField_hypAddToFields</code> .	2433, 2460		
hyp commands:			
<code>\l_hyp_annot_colorfile_bool</code> ...	<a href="#">471</a>		
<code>\l_hyp_annot_colorlink_bool</code> ...	<a href="#">471</a>		
<code>\l_hyp_annot_colormenu_bool</code> ...	<a href="#">471</a>		
<code>\l_hyp_annot_colorrun_bool</code> ....	<a href="#">471</a>		
<code>\l_hyp_annot_colorurl_bool</code> ....	<a href="#">471</a>		
<code>\l_hyp_annot_ocgcolorfile_bool</code> .	<a href="#">475</a>		
<code>\l_hyp_annot_ocgcolorlink_bool</code> .	<a href="#">475</a>		
<code>\l_hyp_annot_ocgcolormenu_bool</code> .	<a href="#">475</a>		
<code>\l_hyp_annot_ocgcolorrun_bool</code> ..	<a href="#">475</a>		
<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	<a href="#">475</a>		
<code>\l_hyp_current_dest_name_tl</code> ....	<a href="#">574</a> , 630, 646		
hyp internal commands:			
<code>\g_hyp_AcroForm_CoFields_prop</code> ..	2394, 2415, 2484, 2486, 2501		
<code>\g_hyp_AcroForm_Fields_prop</code> ...	2395, 2408, 2479		
<code>\l_hyp_annot_GoTo_bool</code> 626, 642, 654			
<code>\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</code> .....	<a href="#">479</a>		
<code>\l_hyp_annot_GoToR_bool</code> .....	705		
<code>\l_hyp_annot_Launch_bool</code> .....	763		
<code>\l_hyp_annot_Named_bool</code> .....	820		
<code>\c_hyp_annot_types_seq</code> .....	<a href="#">434</a> , <a href="#">471</a> , <a href="#">475</a> , <a href="#">938</a> , <a href="#">1235</a> , <a href="#">1333</a>		
<code>\l_hyp_annot_URI_bool</code> .....	667		
<code>\g_hyp_bordercolormodel_str</code> ...	<a href="#">470</a> , 963, 979		
<code>\_hyp_clist_display:n</code> .	103, 110, 119		

\__hyp_color_export:nnN .....	\__hyp_href_url_aux:n .....
..... 37, 851, 851, 867, 979	337, 340
\__hyp_color_export_aux:wnnN ...	\__hyp_href_url_aux:nn .....
..... 851, 855, 862	275, 277
\__hyp_color_select:n 37, 868, 868, 884	\l__hyp_href_url_encode_bool ...
\__hyp_color_select:nN .....	..... 173, 184, 222, 238, 268, 330
37	\__hyp_href_url_format .....
\__hyp_color_select_aux:wn .....	..... 179, 185, 227, 343
..... 868, 872, 879	\l__hyp_href_url_ismap_bool ...
\__hyp_color_set:nN .....	..... 174, 198, 675
38	\l__hyp_href_url_protocol_tl ...
\__hyp_color_set:nn .....	..... 175, 186, 225, 279, 344
..... 38, 371, 373, 885, 885, 901, 943	\__hyp_link_goto_begin:nw .....
\__hyp_color_set_aux:nwn 885, 889, 896	..... 597, 632, 648
\c__hyp_colorscheme_daleif_prop 3585	\__hyp_link_goto_end: . 619, 633, 656
\c__hyp_colorscheme_henryford_-	\c__hyp_map_annot_hyp_prop .... 434
prop .....	\c__hyp_map_hyp_annot_prop .....
3676	..... 434, 902,
\c__hyp_colorscheme_julian_prop 3600	966, 1002, 1029, 1037, 1048, 1075,
\c__hyp_colorscheme_original_-	1083, 1152, 1260, 1296, 1306, 1438
prop .....	\__hyp_ocg_init: 1094, 1094, 1150, 1160
3570	\l__hyp_page/Trans .....
\c__hyp_colorscheme_phelype_prop	498
..... 3661	\__hyp_PageLabels_gpush: 532, 532, 545
\c__hyp_colorscheme_szabolcsA_-	\l__hyp_para_tmpa_tl 430, 774, 777, 786
prop .....	\l__hyp_para_tmpa_tl\l__hyp_-
3630	text_tmpa_str\g__hyp_text_-
\c__hyp_colorscheme_szabolcsB_-	tmpa_str .....
prop .....	427
3645	\__hyp_ref_check:nn 394, 404, 416, 2538
\c__hyp_colorscheme_tivv_prop . 3615	\__hyp_ref_if_exist:nn . 394, 402, 403
\l__hyp_dest_box .....	\__hyp_ref_if_exist:nnTF .. 406, 2542
29,	\__hyp_ref_label:nn 394, 395, 401, 2537
484, 561, 562, 563, 2046, 2062, 2069	\__hyp_ref_value:nn 394, 417, 421, 2546
\l__hyp_dest_name_tmpa_tl .. 427,	\__hyp_secondoftwoithopt:wnn ...
600, 601, 607, 611, 613, 616, 726, 739	..... 356, 357, 358, 359
\l__hyp_dest_pdfremotestartview_-	\__hyp_setup_info_date_key:nn ...
tl .....	..... 2153, 2185, 2186
453, 734, 1853, 1857	\__hyp_setup_info_key:nn .. 2097,
\g__hyp_dest_pdfstartpage_tl ...	2146, 2147, 2148, 2149, 2151, 2152
..... 453, 1864, 1866, 1873, 1891, 1898	\__hyp_store_metadata:nn .....
\g__hyp_dest_pdfstartview_tl ...	..... 360, 365, 2094,
453, 1866, 1873, 1884, 1888, 1891, 1898	2114, 2132, 2166, 2180, 2201, 2273
\l__hyp_dest_pdfview_tl .....	\__hyp_text_cleanup:N . 505, 505, 519
..... 469, 579, 588,	\l__hyp_text_enc_dest_print_tl ..
2015, 2018, 2025, 2028, 2029, 2030,	..... 456, 725
2031, 2032, 2033, 2038, 2042, 2078	\l__hyp_text_enc_dest_tl .....
\c__hyp_dest_startview_regex ...	..... 29, 456, 555, 612
..... 485, 1851, 1882	\l__hyp_text_enc_file_print_tl ..
\c__hyp_dest_undefined_tl .....	..... 456, 768
..... 433, 606, 607	\l__hyp_text_enc_info_print_tl ..
\__hyp_destination:nn .....	..... 456, 530
..... 29, 548, 548, 579, 588	\l__hyp_text_enc_para_print_tl ..
\l__hyp_filename_tmpa_tl ... 427,	..... 456, 773
708, 710, 716, 717, 722, 769, 770, 786	\l__hyp_text_enc_uri_print_tl ...
\__hyp_href_pdf_aux:nn .....	..... 240,
292, 295	243, 270, 273, 332, 335, 456, 672, 1449
\l__hyp_href_pdf_destination_tl .	
..... 176, 187, 226, 297	
\l__hyp_href_pdf_page_tl 177, 195, 733	
\__hyp_href_run_aux:nn .....	
310, 313	
\l__hyp_href_run_parameter_tl ...	
..... 178, 199, 315	



\_hyp\_text\_pdfstring:nnN .....	hidelink .....	1322
..... 513, 513, 525, 527,	hidelinks .....	1322
530, 553, 610, 670, 723, 766, 771, 1449	hidemenu .....	1322
\_hyp\_text\_pdfstring\_info:nN ...	hiderun .....	1322
..... 528, 528, 2108, 2126, 2136	hideurl .....	1322
\_hyp\_text\_purify:nN . 501, 501, 518	hypertextnames .....	1392
\_hyp\_text\_string\_from\_unicode:nN	link .....	9, 1438
..... 509, 509, 520	linkborder .....	13
\g\_hyp\_text\_tmpa\_str . 432, 521, 523	linkborderstyle .....	13, 1048
\l\_hyp\_text\_tmpa\_str .....	linkcolor .....	938
..... 431, 518, 519, 520, 521	linkfileprefix .....	1392
\l\_hyp\_tmpa\_box 422, 1162, 1175, 1180	linktoc .....	1392
\l\_hyp\_tmpa\_int .....	linktocpage .....	1392
..... 422, 1851,	localanchorname .....	1392
1853, 1882, 1884, 2005, 2006, 2010,	menu .....	9, 1438
2012, 2036, 2043, 2044, 2045, 2054,	menuborder .....	13
2056, 2066, 2071, 2488, 2490, 2496	menuborderstyle .....	13, 1048
\l\_hyp\_tmpa\_str .....	menucolor .....	938
..... 422, 2108, 2109, 2111,	naturalnames .....	1392
2126, 2127, 2129, 2136, 2137, 2141	nesting .....	13
\l\_hyp\_tmpa\_tl 422, 556, 560, 568,	ocgcolorfile .....	1188
979, 983, 1449, 1450, 1455, 1849,	ocgcolorlink .....	1188
1851, 1880, 1882, 2012, 2013, 2020	ocgcolorlinks .....	1188
\l\_hyp\_uri\_tmpa\_tl ... 427, 673, 674	ocgcolormenu .....	1188
hyp/anchor .....	ocgcolorrun .....	1188
..... 574	ocgcolorurl .....	1188
hyp/annot/file (color name) .....	pageanchor .....	1392
..... 470	pdfauthor .....	2097
hyp/annot/link (color name) .....	pdfborder .....	13
..... 470	pdfborderstyle .....	13, 1048
hyp/annot/menu (color name) .....	pdfcreationdate .....	13, 2153
..... 470	pdfcreator .....	2097
hyp/annot/run (color name) .....	pdfencoding .....	1360
..... 470	pdfinfo .....	2215
hyp/text/pdfstring .....	pdfkeywords .....	2097
..... 500	pdflang .....	13, 2083
\hypercalcbp .....	pdflinkmargin .....	13
..... 11, 17, 139	pdfmoddate .....	13, 2153
\HyperDestNameFilter .....	pdfproducer .....	2097
..... 554, 611	pdfremotestartview .....	11
\hypersetup 1, 2, 5, 8, 9, 12, 17, 64, 97, 158	pdfstartview .....	11
\hypersetup keys:	pdfsubject .....	2097
allcolors .....	pdftitle .....	2097
..... 938	pdftrapped .....	2187
bookmarkstyle .....	pdfversion .....	1360
..... 12	pdfview .....	11, 2001
bordercolormodel .....	plainpages .....	1392
..... 12, 959	run .....	9, 1438
colorfile .....	runborder .....	13
..... 938	runborderstyle .....	13, 1048
colorlink .....	runcolor .....	938
..... 938	unicode .....	1360
colorlinks .....	url .....	9, 1438
..... 924	urlborder .....	13
colormenu .....		
..... 938		
colormenu .....		
..... 938		
colorrund .....		
..... 938		
colorscheme .....		
..... 1, 1346		
colorurl .....		
..... 938		
debug .....		
..... 1370		
destlabel .....		
..... 12		
draft .....		
..... 1370		
extension .....		
..... 12, 1392		
file .....		
..... 9, 1438		
fileborderstyle .....		
..... 13, 1048		
filecolor .....		
..... 938		
final .....		
..... 1370		
hidefile .....		
..... 1322		

urlborderstyle	13, 1048	
urlcolor	938	
verbose	1370	
hypertextnames (hypersetup key)	1392	
hypListbox internal commands:		
\_hypListbox	2655, 2714	
hypRadio internal commands:		
\_hypRadio	2640, 2662	
hypwritetorep internal commands:		
\_hypwritetorep	3417	
<b>I</b>		
\ifcase	3201, 3325, 3360	
\ifcsname	2978	
\ifdim	2605, 2644, 2645	
\ifnum	3032, 3201, 3207, 3238, 3325, 3330, 3360, 3365, 3490, 3508, 3523	
\iftrue	3492	
\ifx	2437, 2574, 2634, 3009, 3021, 3025, 3042, 3046, 3050, 3054, 3058, 3062, 3066, 3070, 3074, 3078, 3095, 3099, 3202, 3203, 3211, 3215, 3222, 3242, 3246, 3253, 3254, 3326, 3334, 3361, 3369, 3442, 3502	
\immediate	2421, 2424	
int commands:		
\int_compare:nNnTF	2010, 2036, 2706, 3124, 3177, 3287, 3309	
\int_compare_p:nNn	3170, 3278	
\int_eval:n	732	
\int_max:nn	733	
\int_new:N	425	
iow commands:		
\iow_newline:	2927, 2933, 2946	
<b>K</b>		
\kern	3449	
keys commands:		
\keys_define:nn	164, 182, 220, 369, 924, 940, 947, 959, 968, 989, 1004, 1023, 1050, 1069, 1192, 1206, 1218, 1241, 1247, 1262, 1291, 1322, 1335, 1346, 1360, 1370, 1379, 1392, 1419, 1440, 1445, 2001, 2083, 2099, 2117, 2155, 2169, 2187, 2215, 2271, 2286, 2299, 2319	
\l_keys_key_str	169, 2141, 2384	
\keys_set:nn	161, 190, 209, 236, 266, 290, 308, 328, 377, 1352, 1359, 2192, 2219, 2222, 2223, 2224, 2225, 2280, 2309, 2384	
\keys_set_known:nn	2387	
\kvsetkeys	160, 2475	
<b>L</b>		
\label	8, 12	
\LayoutCheckField	2867	
\LayoutChoiceField	2637	
\LayoutPushButtonField	2743, 2760	
\LayoutTextField	2575	
\leavevmode	2576, 2673, 3447	
legacy commands:		
\legacy_if:nTF	146, 386, 390, 540, 2513, 2739, 2780, 2830, 2863, 3147	
\let	237, 267, 291, 309, 329, 357, 358, 359, 637, 688, 695, 749, 755, 799, 805, 848, 2397, 2398, 2400, 2401, 2428, 2524, 2525, 2526, 2557, 2558, 2592, 2593, 2635, 2974, 2981, 2983, 2988, 3005, 3039, 3456, 3470, 3473, 3482, 3493, 3500, 3503, 3505, 3513, 3521	
link (hypersetup key)	9, 1438	
linkborder (hypersetup key)	13	
linkborderstyle (hypersetup key)	13, 1048	
linkcolor (hypersetup key)	938	
linkfileprefix (hypersetup key)	1392	
linktoc (hypersetup key)	1392	
linktocpage (hypersetup key)	1392	
localanchorname (hypersetup key)	1392	
<b>M</b>		
\MakeButtonField	2748, 2763, 2787, 2834, 2845	
\MakeChoiceField	2723	
\MakeFieldObject	2388	
\MakeRadioField	2705	
\MakeTextField	2584	
\mbox	1172	
menu (hypersetup key)	9, 1438	
menuborder (hypersetup key)	13	
menuborderstyle (hypersetup key)	13, 1048	
menucolor (hypersetup key)	938	
mode commands:		
\mode_if_horizontal:TF	550, 572	
\mode_leave_vertical:	233, 263, 287, 305, 325, 599, 682, 741, 788, 825, 2717, 2745, 2762, 2784, 2826, 2869	
msg commands:		
\msg_error:nn	385, 2759, 2844	
\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	148, 1366	
\msg_warning:nnn	157, 603, 843	

<code>\msg_warning:nnnn</code> .....	<code>\pdf_version:</code> .....
1210, 1251, 1488, 1532,	1536, 1655, 1684, 1718, 1733,
1566, 1589, 1651, 1680, 1714, 1729,	1759, 1784, 1817, 1833, 1938, 1965
1742, 1755, 1780, 1813, 1829, 1856,	<code>\pdf_version_compare:NnTF</code> .....
1887, 1920, 1934, 1961, 1995, 2077	1526, 1617, 1639, 1675, 1708, 1723,
<code>\msg_warning:nnnnn</code> .....	1749, 1774, 1800, 1823, 1928, 1955
168,	<code>\pdf_version_compare_p:Nn</code> .....
1282, 1316, 1427, 1508, 1545, 1632,	779, 1189, 1238
1665, 1690, 1766, 1793, 1842, 1947,	<code>\pdf_version_major:</code> .....
1974, 2209, 2326, 2345, 2364, 2375	145, 1190, 1213, 1239, 1255
 <b>N</b>	
<code>naturalnames</code> (hypersetup key) .....	<code>\pdf_version_minor:</code> .. 144, 1213, 1255
<code>nesting</code> (hypersetup key) .....	pdfannot commands:
13	<code>\pdfannot_box:nnnn</code> .....
<code>\newcommand</code> .....	2579, 2690, 2749, 2800, 2835, 2899
142, 3421, 3423, 3425	<code>\pdfannot_box_ref_last:</code> .. 2435, 2451
<code>\newcount</code> .....	<code>\pdfannot_dict_put:nnn</code> .....
2529, 2713	683, 743, 789, 828,
<code>\NewDocumentCommand</code> .....	980, 1015, 1039, 1061, 1085, 1267, 1298
2388	<code>\pdfannot_dict_remove:nn</code> .....
<code>\NewExpandableDocumentCommand</code> ....	974, 1010, 1031, 1056, 1077, 1275, 1308
356	<code>\pdfannot_link:nnn</code> . 684, 744, 790, 829
<code>\newlength</code> .....	<code>\pdfannot_link_goto_begin:nw</code> .. 616
3440	<code>\pdfannot_link_goto_end:</code> .....
<code>\noexpand</code> ....	621
2611, 2612, 2613, 2614, 2992	<code>\pdfannot_link_margin:n</code> .... 8, 1570
<code>\nolinkurl</code> .....	<code>\c_pdfannot_link_types_seq</code> .... 479
3, 4	pdfauthor (hypersetup key) .....
 <b>O</b>	
<code>ocgcolorfile</code> (hypersetup key) .....	2097
1188	<code>\pdfbookmark</code> .....
<code>ocgcolorlink</code> (hypersetup key) .....	3426
1188	pdfborder (hypersetup key) .....
<code>ocgcolorlinks</code> (hypersetup key) ....	13, 1048
1188	pdfborderstyle (hypersetup key) . 13, 1048
<code>ocgcolormenu</code> (hypersetup key) .....	pdfcreationdate (hypersetup key) 13, 2153
1188	pdfcreator (hypersetup key) .....
<code>ocgcolorrun</code> (hypersetup key) .....	2097
1188	<code>\pdfdest</code> .....
<code>ocgcolorurl</code> (hypersetup key) .....	11
 <b>P</b>	
<code>pageanchor</code> (hypersetup key) .....	pdfdict commands:
1392	<code>\pdfdict_new:n</code> 498, 661, 698, 757, 814
<code>\paperwidth</code> .....	<code>\pdfdict_put:nnn</code> .....
3	207, 214, 499, 662, 663, 674,
<code>\PassOptionsToPackage</code> ....	677, 699, 700, 712, 719, 729, 739,
392, 1384, 1389	758, 759, 770, 785, 815, 816, 826,
pdf commands:	1596, 1597, 1601, 1602, 2323, 2333,
<code>\pdf_bdcobject:nn</code> .....	2337, 2340, 2342, 2361, 2370, 2372
1174, 1177	<code>\pdfdict_remove:nn</code> 204, 782, 1606, 1607
<code>\pdf_destination:nn</code> ....	<code>\pdfdict_use:n</code> .....
25, 134, 567	683, 742, 789, 795, 828, 833, 2312
<code>\pdf_destination:nnnn</code> .....	pdfencoding (hypersetup key) .....
560	1360
<code>\pdf_emc:</code> .....	<code>\pdfescapestring</code> .....
1176, 1182	2981
<code>\pdf_link_user:nnn</code> .....	pdffile commands:
2720	<code>\pdffile_embed_file:nnn</code> .....
<code>\pdf_name_from_unicode_e:n</code> .....	714
215, 712, 827	PDFForm internal commands:
<code>\pdf_object_if_exist:nTF</code> .....	<code>\PDFForm_hypName</code> .....
710	3020, 3023, 3027, 3030
<code>\pdf_object_new:nn</code> .....	pdfinfo (hypersetup key) .....
1096,	2215
1097, 1098, 1099, 2914, 2915, 2916	pdfkeywords (hypersetup key) .....
<code>\pdf_object_ref:n</code> .....	2097
135, 722,	pdflang (hypersetup key) .....
1102, 1104, 1126, 1127, 1130, 1134,	13, 2083
1139, 1144, 1149, 2506, 2508, 2972	pdflinkmargin (hypersetup key) .....
<code>\pdf_object_ref_last:</code> ....	13
743, 2314	
<code>\pdf_object_unnamed_write:nn</code> ....	
742, 2310	
<code>\pdf_object_write:nn</code> .....	
1100,	
1106, 1116, 1128, 2917, 2959, 2966	
<code>\pdf_pageobject_ref:n</code> 1873, 1898, 2544	
<code>\pdf_string_from_unicode:nnN</code> ..	
511	



<code>\seq_sort:Nn</code> .....	2490	<code>\@currValue</code> .....	2670, 2700
<code>\setbox</code> .....	350	<code>\@currentHref</code> .....	
<code>\setlength</code> .....	3459, 3530	.....	3463, 3476, 3487, 3517, 3534
<code>\setpdflinkmargin</code> .....	8	<code>\@curropt</code> ..	2589, 2599, 2602, 2666, 2669
<code>\settowidth</code> .....	2604	<code>\@empty</code> .....	2558, 2574, 3042,
<code>\show</code> .....	2664	3046, 3050, 3054, 3058, 3062, 3066,	
sort commands:		3070, 3074, 3078, 3222, 3253, 3254	
<code>\sort_return_same:</code> .....	2494	<code>\@endForm</code> .....	2524
<code>\sort_return_swapped:</code> .....	2493	<code>\@esphack</code> .....	399
<code>\space</code> .....	412, 2672, 3204, 3221,	<code>\@firstofone</code> ..	2428, 2983, 2988, 3005
3222, 3252, 3253, 3327, 3359, 3362		<code>\@firstoftwo</code> .....	3524
<code>\spacefactor</code> .....	550, 572	<code>\@for</code> .....	2600, 2667
<code>\stockwidth</code> .....	3	<code>\@gobbletwo</code> .....	3473
str commands:		<code>\@hyper@launch</code> .....	35, 809
<code>\c_colon_str</code> .....	809	<code>\@ifnextchar</code> .....	245
<code>\c_hash_str</code> .....	679	<code>\@ifundefined</code> .....	2997
<code>\c_percent_str</code> .....	680	<code>\@latex@warning</code> .....	409
<code>\str_case:nnTF</code> .....	3384	<code>\@mainaux</code> .....	2421, 2424, 2444, 2463
<code>\str_case_e:nnTF</code> .....	2006	<code>\@mkboth</code> .....	3473
<code>\str_compare:nNnTF</code> .....	2492	<code>\@nil</code> .....	2995, 3008, 3014
<code>\str_gset:Nn</code> .....	963	<code>\@part</code> .....	3500, 3507
<code>\str_gset_eq:NN</code> .....	521	<code>\@pdfauthor</code> .....	21
<code>\str_head:n</code> .....	2198, 2203	<code>\@pdfborder</code> .....	140
<code>\str_if_eq:nnTF</code> .....		<code>\@pdfborderstyle</code> .....	141
... 557, 2109, 2127, 2137, 2277, 3086		<code>\@savsf</code> .....	550, 572
<code>\str_if_eq_p:nn</code> .....	777, 1190, 1239	<code>\@schapter</code> .....	3470, 3471
<code>\str_lowercase:n</code> .....		<code>\@secondoftwo</code> .....	3526
..... 2006, 2132, 2180, 2199, 2204		<code>\@sect</code> .....	3521, 3522
<code>\str_new:N</code> .....	426, 431, 432, 470	<code>\@spart</code> .....	3513, 3514
<code>\str_set:Nn</code> .....	503	<code>\@ssect</code> .....	3456, 3457
<code>\str_set_convert:Nnnn</code> .....	8	<code>\@tempdima</code> ..	2598, 2605, 2607, 2644,
<code>\str_set_eq:NN</code> .....	523	2645, 2646, 2650, 2651, 2652, 2653	
<code>\str_tail:n</code> .....	2199, 2204	<code>\@tempdimb</code> .....	2604, 2605
<code>\str_uppercase:n</code> .....	2192, 2198, 2203	<code>\@typeset@protect</code> .....	3442
<code>\string</code> .....	2422,	<code>\@unexpandable@protect</code> .....	
2425, 2446, 2464, 2997, 3001, 3012		.....	237, 267, 291, 309, 329
<code>\subpdfbookmark</code> .....	3422	<code>\c@secnumdepth</code> .....	3490, 3508, 3523
sys commands:		<code>\calc@bm@number</code> .....	3434
<code>\c_sys_backend_str</code> .....	3384	<code>\check@bm@number</code> .....	3433
		<code>\Fld@additionalactions</code> ..	3084, 3154,
		3188, 3224, 3261, 3308, 3348, 3377	
		<code>\Fld@align</code> .....	3113, 3199, 3274
		<code>\Fld@altname</code> ..	3021, 3023, 3095, 3097
		<code>\Fld@annotflags</code> .....	3108,
		3161, 3194, 3230, 3269, 3318, 3353	
		<code>\Fld@annotnames</code> .....	3093, 3110,
		3163, 3196, 3232, 3271, 3320, 3355	
		<code>\Fld@bcolor</code> .....	3132, 3134, 3215,
		3217, 3246, 3248, 3282, 3295, 3297	
		<code>\fld@bcolor</code> .....	3203
		<code>\Fld@bordercolor</code> .....	3128,
		3130, 3172, 3181, 3183, 3202, 3211,	
		3213, 3242, 3244, 3280, 3291, 3293,	
		3326, 3334, 3336, 3361, 3369, 3371	

## T

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

<code>\@BOOKMARK</code> .....	3428
<code>\@CheckBox</code> .....	2850
<code>\@ChoiceMenu</code> .....	2590
<code>\@Form</code> .....	2473
<code>\@PushButton</code> .....	2728
<code>\@Reset</code> .....	2817
<code>\@Submit</code> .....	2769
<code>\@TextField</code> .....	2555
<code>\@bookmarksopenlevel</code> .....	1471
<code>\@bsphack</code> .....	397
<code>\@chapter</code> .....	3482, 3483
<code>\@currDisplay</code> .....	2604, 2672

\Fld@borderstyle .....	3114,	\G@refundefinedtrue .....	408
3167, 3200, 3236, 3275, 3324, 3375		\H@old@part .....	3500, 3511
\Fld@borderwidth .	2651, 2652, 3114,	\H@old@chapter .....	3470, 3479
3167, 3200, 3236, 3275, 3324, 3375		\H@old@sect .....	3521, 3532, 3541
\Fld@calculate@code .	2437, 3054, 3056	\H@old@spart .....	3513, 3519
\Fld@calculate@sortkey .....	2448	\H@old@sssect .....	3456, 3461
\Fld@cbsymbol .....	3136	\href@ .....	245, 251
\Fld@charsize .....		\href@split .....	251, 252
.. 2650, 3140, 3221, 3252, 3303, 3359		\Hy@abspage .....	544
\Fld@checkequals .....	2602, 2669	\Hy@activeanchorfalse .....	595
\Fld@choices .....	3223	\Hy@activeanchortrue .....	586
\Fld@color 3141, 3143, 3222, 3253, 3304		\Hy@AtBeginDocument .....	2402, 2418
\Fld@default .....		\Hy@bookmarksnumberedfalse ...	1464
... 2557, 2574, 2592, 2635, 2664,		\Hy@bookmarksnumberedtrue ...	1465
2665, 2853, 3254, 3258, 3259, 3306		\Hy@bookmarksopenfalse .....	1468
\Fld@flags .....	3112,	\Hy@bookmarksopentrue .....	1469
3165, 3198, 3234, 3273, 3322, 3357		\Hy@bookmarkstype .....	1472
\Fld@format@code .....	3046, 3048	\Hy@chapapp .....	3474, 3485
\Fld@height .....		\Hy@colorlink .....	32
2560, 2581, 2584, 2595, 2653, 2692,		\Hy@currentbookmarklevel .....	3415
2705, 2723, 2772, 2820, 2856, 2901		\Hy@DisableOption .....	150
\Fld@keystroke@code .....	3042, 3044	\Hy@drafttrue .....	1383
\Fld@listcount 2663, 2671, 2706, 2713		\Hy@escapeform .....	
\Fld@mappingname 3025, 3027, 3099, 3101		..... 2578, 2675, 2719, 2747,	
\Fld@maxlen .....	3309, 3311	2786, 2833, 2871, 2979, 2986, 2989	
\Fld@menulength ....	2597, 2603, 2650	\Hy@escapestring .....	2981,
\Fld@name .....		2983, 2988, 2991, 2992, 2995, 2996,	
.. 2556, 2591, 2729, 2852, 3020, 3094		3005, 3008, 3044, 3048, 3052, 3056,	
\Fld@onblur@code .....	3062, 3064	3060, 3064, 3068, 3072, 3076, 3080,	
\Fld@onclick@code .....	3187	3136, 3187, 3250, 3306, 3307, 3344	
\Fld@onenter@code .....	3074, 3076	\Hy@finaltrue .....	1388
\Fld@onexit@code .....	3078, 3080	\Hy@FormObjects ....	2478, 2912, 2974
\Fld@onfocus@code .....	3058, 3060	\Hy@gtemp .....	3037, 3039
\Fld@onmousedown@code ...	3066, 3068	\Hy@href .....	245
\Fld@onmouseup@code .....	3070, 3072	\Hy@href@nextactionraw .....	208
\Fld@pageobjref .....		\Hy@href@page .....	196
..... 2526, 2540, 2977, 3109,		\Hy@hypertextnamesfalse .....	1398
3162, 3195, 3231, 3270, 3319, 3354		\Hy@hypertextnametrue .....	1397
\Fld@radiosymbol .....	3250	\Hy@linkfileprefix .....	1400
\Fld@rotation .....	3124,	\Hy@linktoc .....	1423
3126, 3170, 3177, 3179, 3201, 3207,		\Hy@localanchornamefalse ....	1403
3209, 3238, 3240, 3278, 3287, 3289,		\Hy@localanchorname>true .....	1402
3325, 3330, 3332, 3360, 3365, 3367		\Hy@MakeCurrentHrefAuto .....	
\Fld@submitflags .....	3346	..... 3458, 3474, 3485, 3515, 3529	
\Fld@validate@code .....	3050, 3052	\Hy@naturalnamesfalse .....	1407
\Fld@value .....		\Hy@naturalnametrue .....	1406
.. 2558, 2574, 2593, 2634, 2635, 3307		\Hy@next .....	3484, 3493, 3496
\Fld@width .....	2559,	\Hy@numberline .....	143, 3416
2573, 2580, 2584, 2594, 2633, 2644,		\Hy@org@chapter .....	3482, 3497
2646, 2691, 2705, 2723, 2741, 2771,		\Hy@OutlineName .....	3432
2782, 2819, 2831, 2855, 2865, 2900		\Hy@OutlineRerunCheck .....	3430
\Fld@X@additionalactions .....		\Hy@pageanchorfalse .....	1411
..... 3041, 3086, 3089		\Hy@pageanchortrue .....	1410
\Form@action .....	3344	\Hy@pdfmajorversion .....	145



\Hy@pdfminorversion	144	\hyper@anchorend	574, 3451, 3476, 3487, 3517
\Hy@pdfstringtrue	27, 516	\hyper@anchorstart	574, 3451, 3476, 3487, 3517
\Hy@pdfversion	3032	\hyper@link	31, 32, 624
\Hy@plainpagesfalse	1415	\hyper@linkend	31, 32, 652
\Hy@plainpagetrue	1414	\hyper@linkfile	297, 703
\Hy@pstringdef	526, 3012	\hyper@linklaunch	34, 315, 761, 812
\Hy@PutCatalog	532	\hyper@linknamed	36, 153, 818
\Hy@raisedlink	3450, 3475, 3486, 3516	\hyper@linkstart	31, 32, 640
\Hy@RestoreLastskip	571	\hyper@linkurl	279, 343, 665
\Hy@safe@activetrue	681, 2440	\hyper@normalise	245, 275, 292, 310, 337, 350
\Hy@SaveLastskip	551	\HyPL@Labels	534, 544
\Hy@secnum@part	3503, 3505, 3508	\HyPL@storePageLabel	532
\Hy@SectionAnchorHref	3441, 3463, 3534	\HyPsd@SanitizeForOutFile	3419
\Hy@SectionHShift	3440, 3449, 3459, 3530	\if@filesw	2420, 2443, 2462
\Hy@StepCount	2603, 2671	\ifFld@combo	2618, 2648
\Hy@temp	2441, 2456, 2994, 2995	\ifFld@hidden	2573, 2633
\Hy@unicodefalse	3034	\ifFld@multiline	2561
\Hy@VerboseAnchor	552	\ifFld@popdown	2619
\Hy@verbosefalse	1374	\ifFld@radio	2615, 2638
\Hy@VerboseLinkStart	628, 644	\ifHy@implicit	3436
\Hy@VerboseLinkStop	635, 658, 691, 751, 801, 839	\ifHy@pdfescapeform	2980, 2990
\Hy@verbosetrue	1373	\ifHy@unicode	3033
\Hy@VersionChecked	136	\kv@set@family@handler	156
\Hy@WrapperDef	548	\ltx@empty	2397, 2398, 2437, 2524, 2525, 2526, 2557
\Hy@xspace@end	632, 690, 750, 800, 838	\ltx@gobble	2400
\HyAnn@AbsPageLabel	2525, 2531, 2577, 2674, 2718, 2746, 2785, 2832, 2870	\ltx@gobbletwo	2401
\HyAnn@Count	2529, 2530, 2533, 2534, 2535, 2536, 2537, 2538, 2542, 2546	\ltx@ifUndefined	3481, 3491
\HyField@AddToFields	2431, 2585, 2707, 2724, 2755, 2811, 2840, 2904	\ltx@one	2533
\HyField@afields	2397	\ltx@ReturnAfterFi	3013
\HyField@AfterAuxOpen	2402, 2428, 2461	\ltx@zero	2530
\HyField@AuxAddToCoFields	2401, 2413, 2425, 2446	\m@ne	3490, 3505
\HyField@AuxAddToFields	2400, 2406, 2422, 2464	\OBJ@OCG@view	142
\HyField@cofields	2398	\p@	2607
\HyField@FlagsCheckBox	2862	\pdf@ifdraftmode	2476
\HyField@FlagsChoice	2643	\PDF@SetupDoc	137
\HyField@FlagsPushButton	2738, 2778, 2829	\PDFForm@Check	2871, 2903, 3105
\HyField@FlagsRadioButton	2639	\PDFForm@List	2719, 2722, 3192
\HyField@FlagsSubmit	2779	\PDFForm@Name	2571, 2632, 2735, 2861, 3019
\HyField@FlagsText	2572	\PDFForm@Push	2747, 2753, 3158
\HyField@PDFChoices	2716	\PDFForm@Radio	2675, 2695, 3227
\HyField@SetKeys	2568, 2609, 2612, 2631, 2731, 2774, 2822, 2857	\PDFForm@Reset	2833, 2839, 3351
\HyPat@ObjRef	3388, 3395, 3402, 3409	\PDFForm@Submit	2786, 2805, 3316
\hyper@@link	253	\PDFForm@Text	2578, 2583, 3266
\hyper@anchor	574	\protected@edef	600
		\strip@pt	3140, 3221, 3252, 3303, 3359
		\toks@	3461, 3467, 3532, 3538
		\tw@	3462, 3467, 3533, 3538
		\url@	350, 353
		\Url@def	351

