

The hyperref-generic module

A generic driver for hyperref

The L^AT_EX Project*

Version 0.96e, released 2024-02-22

This module generates a generic driver for `hyperref` meant to be used with the new L^AT_EX 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 (this requires L^AT_EX-2022-06-01 or newer):

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

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

1 Avoiding transition problems

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

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

*E-mail: latex-team@latex-project.org

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

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

The current handling of the metadata is problematic:

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

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

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

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

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

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

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

```
\cs{DocumentMetadata}{lang=de-DE}
```

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

(Expandable) commands are expanded and special chars can also be input by commands but beside this no conversion is done: for all input `hyperref` basically assumes that the input is already a valid percent encoded url or a valid file name. `hyperref` also doesn't extend or add protocols.

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.

- Citations have by default no special color, they are colored like other internal links. You can use `citecolor` and `citebordercolor` to assign them a special color. This color is not reset if you use `allcolors` or switch to another color scheme. If you want the colors to follow `linkcolor` again you should remove the label `hyp/cite` and/or `hyp/citeborder` from the hook `hyp/link/cite`.
- a number of color schemes have been predefined.

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

- Beside `pdfborder` there are also `linkborder`, `urlborder` etc

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

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

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

The default is `phelype`.

7.3 Public interfaces

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

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

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

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

link margin The driver sets a default link margin—this is identical to `pdftex` and `luatex` driver, but a change for the `xetex` and `dvips` driver. The (undocumented) command `\setpdflinkmargin` does nothing. Use either the key `pdflinkmargin` or `\pdfannot_link_margin:n` to change the margin. See also the description in section 14 and in the `hyperref` manual.

8 PDF strings

`hyperref` uses a command called `\pdfstringdef` to convert text input into something that makes sense and is valid in a PDF string, e.g. in the bookmarks or in the info dictionary or as form field values.

As the handling of the outlines are delegated to the `bookmark` package, they will for now still use `\pdfstringdef`, but all other strings produced by this driver will use a new method based on the `expl3` commands `\text_purify:n` and `\str_set_convert:Nnnn`. For normal text it shouldn't matter, but a variety of commands and math are handled differently. Like with `\pdfstringdef` they are a number of ways to adjust the outcome of `\text_purify:n`. These are described in the `expl3` documentation `interface3.pdf`.

The new method is under heavy development!

Important differences here are

- *This new method requires that files are utf8-encoded* (at least if non-ascii chars are used in for PDF strings).
- All robust commands are currently removed, unless an equivalent has been declared.
- Currently the new method is much more silent: it doesn't warn like `hyperref` if it removes commands.

9 Package options from hyperref

The driver will process the package options at the end. But normally options should better be set with `\hypersetup` after the package has been loaded. This is also the case for options which normally don't work in `\hypersetup`. One option that currently doesn't work correctly as package option is `ocgcolorlinks`

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `destlabel` (destination names are taken from `\label` if possible)
- `encap`
- `hyperfigures` (according to the `hyperref` manual it makes figures hyper links, but actually is a no-op for most drivers, and it does nothing with this driver either.)
- `hyperfootnotes`
- `hyperindex`
- `implicit` (redefine `LATEX` internals)
- `nesting` unneeded key, see comment below in 14. At some time it will be either removed or extended (if some use can be found).
- `pagebackref`
- `pdfpagelabels` (set PDF page labels)
- `psdextra` this loads some extra definitions used by `\pdfstringdef`. The new driver uses `\pdfstringdef` only for the bookmarks, for other strings it is not relevant.

Options that can be without problems set as package options are

- `debug`, `verbose` (a boolean)
- `bookmarks` (a boolean)
- `plainpages`
- `draft`, `final`
- `hypertextnames`
- `naturalnames`
- `pageanchor`

Ignored options:

- All driver options like `pdftex`, `dvipdfmx`, ...
- `raiselinks` (only used in the `dviwind`, `textures` and `tex4ht` driver anyway)
- `frenchlinks`
- `setpagesize`
- `addtopdfcreator`

10 Disabling links

`hyperref` knows like many packages the options `draft` and `final`. With `hyperref` they can be used as package options or in the preamble in `\hypersetup` and disable links and anchors completely. The new driver passes the options also to the `bookmark` package if `bookmark` hasn't been loaded yet as bookmarks can't work properly if the anchors from `hyperref` are missing.

`link` (*setup key*) The `draft` option is a global option that can't be undone (at least not easily). So the new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference without link (this is naturally also possible with `\ref*{abc}`). This disables also all hooks of the link type, so the link is for example no longer colored. It also removes the implicit grouping of the content.

`nested-links` (*setup key*)

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

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

pdfversion The `pdfversion` should be set in `\DocumentMetadata`

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

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

unicode This is always true.

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

```
\DocumentMetadata{pdfstandard=A-2b}
```

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

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

13 Destinations

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

From these instructions two (`Fit` and `FitB`) don't take an argument. All others take one (`FitH`, `FitV`, `FitBH`, `FitBV`) or more (`XYZ`, `FitR`) arguments. These arguments are normally coordinates, `XYZ` takes also a zoom factor. The coordinates are absolute coordinates in `bp` relative to the lower left corner of the PDF.

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

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

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

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

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

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

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

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

<code>nesting</code> (<i>setup key</i>)	Typically PDF viewer can handle only GoToR annotations pointing to a PDF. So normally the default value <code>pdf</code> 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.		
<code>pdfborder</code> (<i>setup key</i>)	This key set accept as value three numbers or three numbers and an array describing		
<code>linkborder</code> (<i>setup key</i>)	a dash pattern, examples are <code>0 0 1</code> or <code>0 0 1 [3 2]</code> . The first two numbers should		
<code>urlborder</code> (<i>setup key</i>)	according to the reference set round corners, but PDF viewer seem to ignore it. The		
<code>runborder</code> (<i>setup key</i>)	third number is the line width of the border. Settings done with <code>pdfborderstyle</code> should		
<code>menuborder</code> (<i>setup key</i>)	take precedence.		
<code>pdfborderstyle</code> (<i>setup key</i>)	The value of this key is the content of the BS dictionary. As an example		
<code>linkborderstyle</code> (<i>setup key</i>)	<code>/Type/Border /W 1 /S/U /D[3 2]</code>		
<code>urlborderstyle</code> (<i>setup key</i>)	Key	Values	comment / example
<code>fileborderstyle</code> (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
<code>runborderstyle</code> (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
<code>menuborderstyle</code> (<i>setup key</i>)	<code>/S</code>	<code>/S</code>	solid (default)
		<code>/D</code>	dash pattern can be set with <code>/D</code>
		<code>/B</code>	beveled
		<code>/I</code>	inset
		<code>/U</code>	underline
	<code>/D</code>	<code><array of numbers></code>	dash pattern (lines/gaps) (default [3])
<code>pdfcreationdate</code> (<i>setup key</i>)	Setting these keys is normally not needed. If they are used the values of the first		
<code>pdfmoddate</code> (<i>setup key</i>)	two keys are stored directly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> .		
<code>pdfmetadate</code> (<i>setup key</i>)	All three keys are used in XMP-metadata. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> (<i>setup key</i>)	As described in the <code>hyperref</code> manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf _l atex and lua _l atex work by page, with dvipdfmx the setting is global (and has to be done in the preamble).		
<code>pdflang</code> (<i>setup key</i>)	The key will work, but it is recommended to the set the language in <code>\DocumentMetadata</code> instead.		

File I

hyperref-generic driver implementation

```

1 \*package>
2 \@@=hyp>
3 \ProvidesFile{hgeneric-testphase.def}[2024-02-22 v0.96e %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?

```

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

```

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

```

1 messages

Redirect the message name:

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

At first a message for the testing of the resource management

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

The pdfversion should be set in \DocumentMetadata

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

A generic message for ignored keys.

```
51 \msg_new:nnn
52 { hyp }
53 { key-dropped }
54 {
55   This~hyperref~driver~ignores~the~key~#1!\\
56   Please~check~the~documentation.
57 }
```

```
57 }
```

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

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

pdf/A message for not allowed Named actions

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

A message if the destination name is empty.

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

A message if the destination check fails

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

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

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



```

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

```

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

```

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

```

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

```

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

```

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

```

149 \msg_new:nnn

```

```

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

```

2 Variants

```

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

```

3 Overwriting/providing commands from hyperref

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

```

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

```

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

\hypercalcbp

```

162 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

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

```

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

```

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

```

\Acrobatmenu should use the new internal link command

```

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

```

`\hypersetup` should set the new keys. We can't also execute `\kvsetkeys{Hyp}` as this errors for example with colors. This means the driver has to provide new code for every key!

```

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

```

Hyperref creates a number of destinations automatically. E.g. in unnumbered chapters and sections and with `\phantomsection`. The following key allows to force a specific name for the destination so that it can be used by bookmarks.

```

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

```

Allow non-ascii in href, and add more href versions. We add a few new keys: `urlencode` to force percent encoding (`\hrefurl`, `\href`) protocol to add a protocol (`\hrefurl`, `\href` doesn't work here as it needs the colon for the split and the guessing.) `destination` to add a destination (`\hrefpdf`)

```

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

```

```

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

```

```

273     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
274   }
275   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
276 }
277
278 \begingroup
279   \catcode'\$=6 %
280   \catcode'\#=12 %
281   \gdef\href@#1{\expandafter\href@split$1##\}%
282   \gdef\href@split$1#$2#$3\\$4{%
283     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
284   \endgroup
285   \hook_use:n{cmd/href/after}
286   }%
287 \endgroup
288
289 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
290
291 \DeclareRobustCommand*{\hrefurl}[1] []
292 {
293   \mode_leave_vertical:
294   \hook_use:n{cmd/href/before}
295   \group_begin:
296   \keys_set:nn { hyp / href } {#1}
297   \bool_if:NTF \l__hyp_href_url_encode_bool
298   {
299     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
300   }
301   {
302     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
303   }
304   \hyper@normalise\__hyp_href_url_aux:nn}
305
306 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
307 {
308   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
309   \group_end:
310   \hook_use:n{cmd/href/after}
311 }
312
313 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
314 \DeclareRobustCommand*{\hrefpdf}[1] []
315 {
316   \mode_leave_vertical:
317   \hook_use:n{cmd/hrefpdf/before}
318   \group_begin:
319   \keys_set:nn { hyp / href } {#1}
320   \hyper@normalise\__hyp_href_pdf_aux:nn
321 }
322
323 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
324 {
325   \exp_args:Nno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
326   \group_end:

```

```

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

```

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

```

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

```

```

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

```

make the new commands compatible with `\pdfstringdef`:

```

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

```

4 Compability commands

4.1 Metadata

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

```

386 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
387 {
388   %\tl_set:cn {@#1}{#2}
389   \AddToDocumentProperties[hyperref]{#1}{#2}
390 }
391 \cs_generate_variant:Nn \__hyp_store_metadata:nn {en,ne,ee}

```

4.2 citecolor

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

```

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

```

```

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

```

5 Checks

The driver can not work properly if the pdfmanagement is not active, as keys need to write to the catalog and to info. But annotations and outlines should work. So should this be a fatal error? Should there be a difference between missing and inactive management? TODO

```

424 \bool_lazy_and:nnF
425 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
426 { \msg_error:nn { hyp}{ missing-resource-management } }

```

Outlines/bookmarks require the bookmark package. TODO check pdfpagemode if bookmarks are suppressed. TODO We overwrite the color key here for now, but this should be moved to bookmark

```

427 \AddToHook { package/bookmark/after}
428 {
429     \define@key{BKM}{color}
430     {
431         \__hyp_color_set:ne {__hyp/tmpa}{#1}
432         \color_export:nVN
433         {__hyp/tmpa}
434         \g__hyp_bordercolormodel_str
435         \BKM@color
436     }
437 }
438 \legacy_if:nT { Hy@bookmarks }
439 {
440     \AddToHook{begindocument/before}[hyperref/bookmark]
441     {
442         \RequirePackage{bookmark}
443     }
444 }
445 \legacy_if:nT { Hy@draft}
446 {
447     \PassOptionsToPackage{draft}{bookmark}
448 }

```

6 Reference and label commands

This uses the in-built property module.

```

\__hyp_property_record:nn

```

```

449 %

```


A label command which adds the space commands from LaTeX:

```

450 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
451 {
452   \@bsphack
453   \property_record:nn{#1}{#2}
454   \@esphack
455 }

```

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

```

456 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}

```

(End of definition for __hyp_property_record:nn.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str
457 \box_new:N \l__hyp_tmpa_box
458 \tl_new:N \l__hyp_tmpa_tl
459 \seq_new:N \l__hyp_tmpa_seq
460 \int_new:N \l__hyp_tmpa_int
461 \str_new:N \l__hyp_tmpa_str

```

(End of definition for \l__hyp_tmpa_tl and others.)

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

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
\l__hyp_text_tmpa_str
\l__hyp_text_tmpa_str
462 \tl_new:N \l__hyp_dest_name_tmpa_tl
463 \tl_new:N \l__hyp_uri_tmpa_tl
464 \tl_new:N \l__hyp_filename_tmpa_tl
465 \tl_new:N \l__hyp_para_tmpa_tl
466 \str_new:N \l__hyp_text_tmpa_str
467 \str_new:N \l__hyp_text_tmpa_str

```

TODO: document and check use!

(End of definition for \l__hyp_dest_name_tmpa_tl and others.)

7.2 Constants

This variable is used if a destination name is empty.

```

468 \tl_const:Nn \c__hyp_dest_undefined_tl {UNDEFINED}

```

(End of definition for \c__hyp_dest_undefined_tl.)

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

```

\c__hyp_annot_types_seq
\c__hyp_map_annot_hyp_prop
\c__hyp_map_hyp_annot_prop
469 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
470 {url,link,file,menu,run}
471 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
472 {
473   URI    = url,

```

```

474     GoTo   = link,
475     GoToR  = file,
476     Named  = menu,
477     Launch = run
478   }
479   \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
480   {
481     url    = URI,
482     link   = GoTo,
483     file   = GoToR,
484     menu   = Named,
485     run    = Launch
486   }
487

```

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

7.3 Variables

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

```

488 \tl_new:N \g__hyp_dest_pdfstartpage_tl
489 \tl_new:N \g__hyp_dest_pdfstartview_tl
490 \tl_new:N \l__hyp_dest_pdfremotestartview_tl

```

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

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

```

\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_tl
\l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_file_print_tl
\l__hyp_text_enc_para_print_tl

491 \tl_new:N \l__hyp_text_enc_uri_print_tl
492 \tl_new:N \l__hyp_text_enc_info_print_tl
493 \tl_new:N \l__hyp_text_enc_dest_tl
494 \tl_new:N \l__hyp_text_enc_dest_print_tl
495 \tl_new:N \l__hyp_text_enc_file_print_tl
496 \tl_new:N \l__hyp_text_enc_para_print_tl
497
498 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
499 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
500 \tl_set:Nn \l__hyp_text_enc_dest_tl      {utf8/string-raw}
501 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
502 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
503 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

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

`\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.

```

504 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End of definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

`\g__hyp_bordercolormodel_str`

`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`

```
505 \str_new:N \g__hyp_bordercolormodel_str
```

(End of 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_colorrrun_bool`

`\l_hyp_annot_colormenu_bool`

```
506 \seq_map_inline:Nn \c__hyp_annot_types_seq
```

```
507 {
```

```
508   \bool_new:c {l_hyp_annot_color#1_bool}
```

```
509 }
```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrrun_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.

```
510 \seq_map_inline:Nn \c__hyp_annot_types_seq
```

```
511 {
```

```
512   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
```

```
513 }
```

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

`\not_Named_bool`, `\l_hyp_annot_Launch_bool`

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

```
514 \seq_map_inline:Nn \c_pdfannot_link_types_seq
```

```
515 {
```

```
516   \bool_new:c {l__hyp_annot_#1_bool}
```

```
517   \bool_set_true:c {l__hyp_annot_#1_bool}
```

```
518 }
```

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

7.5 Boxes

`\l__hyp_dest_box`

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

```
519 \box_new:N \l__hyp_dest_box
```

(End of definition for `\l__hyp_dest_box`.)

7.6 Regex

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

```

520 \regex_const:Nn \c__hyp_dest_startview_regex
521 {
522   \A\ *
523   (?:
524     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
525     |
526     (?:Fit\b|FitB\b)
527     |
528     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
529     |
530     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
531   )
532 }
```

(End of definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

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

```

533 \pdfdict_new:n {l__hyp_page/Trans}
534 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for `l__hyp_page/Trans`.)

8 PDF string conversion

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

`hyp/text/pdfstring`

```

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

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

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

`__hyp_text_purify:nN`

```

536 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
537 {
538   \str_set:Ne #2 {\text_purify:n { #1 } }
539 }
```

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

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

`__hyp_text_cleanup:N`

```

540 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
541 {
542
543 }

```

(End of definition for __hyp_text_cleanup:N.)

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

```

544 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
545 {
546   \pdf_string_from_unicode:nVN { #1 } #2 #2
547 }

```

(End of definition for __hyp_text_string_from_unicode:nN.)

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

`__hyp_text_pdfstring:nnN`

```

548 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
549 {
550   \group_begin:
551   \Hy@pdfstringtrue
552   \hook_use:n {hyp/text/pdfstring}
553   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
554   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
555   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
556   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
557   \group_end:
558   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
559 }
560 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {enN,onN,eoN,ooN,noN}

```

(End of definition for __hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

```

561 \cs_new_protected:Npn \Hy@pstringdef #1 #2
562 { \__hyp_text_pdfstring:enN {#2} {utf8/string-raw}#1 }

```

This is a special version for info keys:

```

\__hyp_text_pdfstring_info:nN

```

```

563 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
564 {
565   \__hyp_text_pdfstring:nn { #1 } { \l__hyp_text_enc_info_print_tl } #2
566 }

```

(End of definition for __hyp_text_pdfstring_info:nN.)

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects \thepage and tries to figure out the numbering system and the prefix. E.g. A-30 is correctly split. If the counter can not be identified hyperref generates only /P entries with the whole content.

The new implementation makes use of the pdf management: The relevant entry in the catalog is continuously updated and pushed out at the end of the document. This works (hopefully ...) with all drivers.

We do not try to avoid the (in hyperref’s wording) “useless” pagelabel entry /PageLabels <</Nums[0<</S/D>>]>> (but it would be possible), we also don’t test for empty \thepage, hyperref seems to handle this fine and the pdf is valid.

The code has to define \Hy@PutCatalog as we can’t yet change code in hyperref. The switch for draftmode has been removed.

```

\__hyp_PageLabels_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel

```

```

567 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
568 {
569   \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
570 }
571
572 \def\Hy@PutCatalog #1 {}
573
574
575 \legacy_if:nT { Hy@pdfpagelabels }
576 {

```

```

577 \cs_set_protected:Npn \HyPL@StorePageLabel #1
578 {
579   \tl_gput_right:Ne \HyPL@Labels { \the\Hy@abspage<<#1>> }
580   \__hyp_PageLabels_gpush:
581 }
582 }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Link level

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

`g__hyp_linknestlevel_int`

```

583 \int_new:N \g__hyp_linknestlevel_int

```

(End of definition for `g__hyp_linknestlevel_int`.)

```

584 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
585 {
586   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
587   { \prg_return_false: }
588   { \prg_return_true: }
589 }
590 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
591 {
592   \use_i:nn {#1}{#2}
593 }
594 \keys_define:nn { hyp }
595 {
596   nested-links .choice:,
597   nested-links / true .code:n =
598   { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },

```

```

599     nested-links / false .code:n =
600     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
601     nested-links .default:n = {true}
602 }

```

10.2 Anchors / destinations

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

```

\__hyp_destination:nn \__hyp_destination:nn {(destination name)} {(location)}

```

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

```

\__hyp_destination:nn
603 \Hy@WrapperDef \__hyp_destination:nn #1 #2
604 {
605   \mode_if_horizontal:T { \@savs\spacefactor }
606   \Hy@SaveLastskip      %defined in hyperref
607   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
608   \__hyp_text_pdfstring:eoN
609   { \HyperDestNameFilter{#1} }
610   { \l__hyp_text_enc_dest_tl }
611   \l__hyp_tmpa_tl
612   \str_if_eq:nnTF {#2} {fitrbx}
613   {
614     \exp_args:NV
615     \pdf_destination:nnnn \l__hyp_tmpa_tl
616     { \box_wd:N \l__hyp_dest_box }
617     { \box_ht:N \l__hyp_dest_box }
618     { \box_dp:N \l__hyp_dest_box }
619   }
620   {
621     \exp_args:NV
622     \pdf_destination:nf
623     { \l__hyp_tmpa_tl }
624     { #2 }
625   }
626   \Hy@RestoreLastskip %defined in hyperref
627   \mode_if_horizontal:T { \spacefactor\@savs }
628 }

```

(End of definition for __hyp_destination:nn.)

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


```

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

```

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

10.3 GoTo Links

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

At first the internal link commands:

```

652 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
653 {
654   \mode_leave_vertical:
655   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
656   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
657   {
658     \msg_warning:nne
659       { hyp }
660       { empty-destination-name }
661       { \c__hyp_dest_undefined_tl }
662     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
663   }
664   {
665     \__hyp_text_pdfstring:eoN
666     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
667     { \l__hyp_text_enc_dest_tl }
668     \l__hyp_dest_name_tmpa_tl
669   }
670   \exp_args:No

```

```

671     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
672 }
673
674 \cs_new_protected:Npn \__hyp_link_goto_end:
675 {
676     \pdfannot_link_goto_end:
677 }

```

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

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

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

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

```

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

```

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

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

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

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

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

```

678
679 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
680 {
681     \bool_if:NTF \l__hyp_annot_GoTo_bool
682     {
683         \int_gincr:N \g__hyp_linknestlevel_int
684         \__hyp_check_link_nesting:TF
685         {
686             \Hy@VerboseLinkStart{#1}{#2}
687             \group_begin:
688             \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
689             \hook_use:n {hyp/link/#1}
690             \__hyp_link_goto_begin:nw {#2} #3 \Hy@xspace@end
691             \__hyp_link_goto_end:
692             \group_end:
693             \Hy@VerboseLinkStop
694         }
695         {
696             \group_begin: #3 \group_end:
697         }
698         \int_gdecr:N \g__hyp_linknestlevel_int

```

```

699     }
700     {{\let\protect\relax#3}}
701   }
702   \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
703   {
704     \bool_if:NT \l__hyp_annot_GoTo_bool
705     {
706       \int_gincr:N\g__hyp_linknestlevel_int
707       \__hyp_check_link_nesting:TF
708       {
709         \Hy@VerboseLinkStart{#1}{#2}% only for debug
710         \group_begin:
711         \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
712         \hook_use:n {hyp/link/#1}
713         \__hyp_link_goto_begin:nw {#2}
714       }
715       {
716         \group_begin:
717       }
718     }
719   }
720
721   \cs_new_protected:Npn \hyper@linkend
722   {
723     \bool_if:NT \l__hyp_annot_GoTo_bool
724     {
725       \__hyp_check_link_nesting:TF
726       {
727         \__hyp_link_goto_end:
728         \group_end:
729         \Hy@VerboseLinkStop
730       }
731       {
732         \group_end:
733       }
734       \int_gdecr:N\g__hyp_linknestlevel_int
735     }
736   }

```

10.4 URI links

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

```

737   \pdfdict_new:n {l_hyp/annot/A/URI}
738   \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
739   \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
740
741   \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
742   {
743     \bool_if:NTF \l__hyp_annot_URI_bool
744     {
745       \int_gincr:N\g__hyp_linknestlevel_int
746       \__hyp_check_link_nesting:TF
747       {
748         \group_begin:

```

```

749 \__hyp_text_pdfstring:eoN
750 { #2}
751 { \l__hyp_text_enc_uri_print_tl }
752 \l__hyp_uri_tmpa_tl
753 \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
754 \bool_if:NT \l__hyp_href_url_ismap_bool
755 {
756 \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
757 }
758 \cs_set_eq:NN \# \c_hash_str
759 \cs_set_eq:NN \% \c_percent_str
760 \Hy@safe@activetrue
761 \mode_leave_vertical:
762 \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
763 \pdfannot_link:nen { URI }
764 {
765 }
766 {
767 \let\protect\relax
768 #1
769 \Hy@xspace@end
770 \Hy@VerboseLinkStop %where is the start??
771 }
772 \group_end:
773 }
774 {
775 \group_begin: #1 \group_end:
776 }
777 \int_gdecr:N\g__hyp_linknestlevel_int
778 }
779 {\let\protect\relax#1}}
780 }
781

```

10.5 GoToR Links files

```

782 \pdfdict_new:n {l_hyp/annot/A/GoToR}
783 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
784 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
785
786 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
787 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
788 {
789 \bool_if:NTF \l__hyp_annot_GoToR_bool
790 {
791 \int_gincr:N\g__hyp_linknestlevel_int
792 \__hyp_check_link_nesting:TF
793 {
794 \group_begin:
795 \tl_set:Ne \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
796 \exp_args:Ne
797 \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
798 {
799 \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n

```

```

800         \pdffile_embed_file:noe
801         {}
802         {\l__hyp_filename_tmpa_tl }
803         {__hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl }
804     }
805     \pdfdict_put:nne
806     {l_hyp/annot/A/GoToR}
807     {F}
808     {\pdf_object_ref:e {__hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl}}
809     \__hyp_text_pdfstring:nnN
810     { #3 }
811     { \l__hyp_text_enc_dest_print_tl }
812     \l__hyp_dest_name_tmpa_tl
813     \tl_if_blank:eTF {#3}
814     {
815         \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
816         {
817             [
818                 \int_eval:n
819                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
820                 /\l__hyp_dest_pdfremotestartview_tl
821             ]
822         }
823     }
824     {
825         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
826     }
827     \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>

```

828     \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
829     \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
830     \pdfannot_link:nnn %expansion??
831     { GoToR }
832     {
833     }
834     {
835         \let\protect\relax
836         #1\Hy@xspace@end
837         \Hy@VerboseLinkStop %where is the start??
838     }
839     \group_end:
840 }
841 {
842     \group_begin: #1 \group_end:
843 }
844 \int_gdecr:N\g__hyp_linknestlevel_int
845 }
846 {{\let\protect\relax#1}}
847 }

```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

848 \pdfdict_new:n {l_hyp/annot/A/Launch}
849 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
850 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
851
852 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
853 {
854   \bool_if:NTF \l__hyp_annot_Launch_bool
855   {
856     \int_gincr:N\g__hyp_linknestlevel_int
857     \__hyp_check_link_nesting:TF
858     {
859       \group_begin:
860       \__hyp_text_pdfstring:nnN
861       { #1 }
862       { \l__hyp_text_enc_file_print_tl }
863       \l__hyp_filename_tmpa_tl
864       \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
865       \__hyp_text_pdfstring:noN
866       { #3 }
867       { \l__hyp_text_enc_para_print_tl }
868       \l__hyp_para_tmpa_tl
869       \bool_if:nTF
870       {
871         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}{}
872         ||
873         \pdf_version_compare_p:Nn > {1.9}
874       }
875       {
876         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
877       }
878       {
879         \pdfdict_put:nne
880         {l_hyp/annot/A/Launch}
881         {Win}
882         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
883       }
884       \mode_leave_vertical:
885       \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
886       \pdfannot_link:nen
887       { Launch }
888       {
889         % /A
890         % <<
891         % \pdfdict_use:n {l_hyp/annot/A/Launch}
892         % >>
893       }
894       {
895         \let\protect\relax
896         #2\Hy@xspace@end
897         \Hy@VerboseLinkStop %where is the start??
898       }
899       \group_end:
900     }
901     { \group_begin: #2 \group_end: }

```

```

902     \int_gdecr:N\g__hyp_linknestlevel_int
903   }
904   {\let\protect\relax#2}}
905 }

```

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

```

906 \use:e
907 { % filename, anchor text, linkname
908   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
909 }
910 {
911   \hyper@linklaunch {#1}{#2}{#3}
912 }

```

10.7 Named links (menu)

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

```

913 \pdfdict_new:n {l_hyp/annot/A/Named}
914 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
915 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
916
917 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
918 {
919   \bool_if:NTF \l__hyp_annot_Named_bool
920   {
921     \int_gincr:N\g__hyp_linknestlevel_int
922     \__hyp_check_link_nesting:TF
923     {
924       \group_begin:
925       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
926       {
927         \mode_leave_vertical:
928         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
929           {\pdf_name_from_unicode_e:n{#1}}
930         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named
931         \pdfannot_link:nnn { Named }
932         {
933           % /A
934           % <<
935           % \pdfdict_use:n { l_hyp/annot/A/Named }
936           % >>
937         }
938         {
939           #2
940           \Hy@xspace@end
941           \Hy@VerboseLinkStop
942         }
943       }
944       {
945         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
946         #2
947       }
948     }
949   }

```

```

949         }
950         { \group_begin: #2 \group_end: }
951         \int_gdecr:N\g__hyp_linknestlevel_int
952     }
953     {{\let\protect\relax#2}}
954 }
955

```

11 Link decorations

11.1 Functions to export and select colors

We support two input syntax: color expressions and model with values. Exporting can be done by first setting the color with `__hyp_color_set:nn` (if needed to a temporary color name) and then using `\color_export:nnN`. But we need a variant as the export format `space-sep-cmyk` or `space-sep-rgb` is stored in a tl.

```

956 \cs_generate_variant:Nn \color_export:nnN {nVN}

```

```

\__hyp_color_select:n \__hyp_color_select:n {\color}

```

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

```

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

```

```

957 \cs_new_protected:Npn \__hyp_color_select:n #1
958 {
959     \tl_if_head_eq_charcode:nNTF {#1}[ %]
960     {
961         \__hyp_color_select_aux:wn #1
962     }
963     {
964         \color_select:n {#1}
965     }
966 }
967
968 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
969 {
970     \color_select:nn {#1}{#2}
971 }
972
973 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

(End of definition for __hyp_color_select:n and __hyp_color_select_aux:wn.)

```

\__hyp_color_set:nn \__hyp_color_set:nn {\ name } {\color}

```

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

```

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

```



```

974 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
975 {
976   \tl_if_head_eq_charcode:nNTF {#2}[ %]
977   {
978     \__hyp_color_set_aux:nwn { #1 } #2
979   }
980   {
981     \color_set:nn {#1} {#2}
982   }
983 }
984
985 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
986 {
987   \color_set:nnn {#1}{#2}{#3}
988 }
989
990 \cs_generate_variant:Nn \__hyp_color_set:nn {ne}

```

(End of definition for `__hyp_color_set:nn` and `__hyp_color_set_aux:nwn`.)

11.2 Textcolor of links

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

```

991 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
992 {
993   \hook_gput_code:nnn
994     {pdfannot/link/#2/begin}
995     {hyp/color}
996     {
997       \bool_if:cT { l_hyp_annot_color#1_bool }
998       {
999         \group_begin:
1000         \color_select:n { hyp/color/#1}
1001       }
1002     }
1003   \hook_gput_code:nnn
1004     {pdfannot/link/#2/end}
1005     {hyp/color}
1006     {
1007       \bool_if:cT { l_hyp_annot_color#1_bool }
1008       {
1009         \group_end:
1010       }
1011     }
1012 }

```

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

```

1013 \keys_define:nn { hyp }
1014 {
1015   ,colorlinks .meta:n =
1016     {
1017       ,pdfborder={0~0~0}

```

```

1018         ,pdfborderstyle=
1019         ,colorurl  =#1
1020         ,colorlink =#1
1021         ,colorrun  =#1
1022         ,colormenu =#1
1023         ,colorfile =#1
1024     }
1025     ,colorlinks .default:n = {true}
1026 }

colorurl (setup key)
colorlink (setup key) 1027 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrun (setup key) 1028 {
colormenu (setup key) 1029     \keys_define:nn { hyp }
colorfile (setup key) 1030     {
    urlcolor (setup key) 1031         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1032         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
    runcolor (setup key) 1033     }
    menucolor (setup key) 1034 }
filecolor (setup key) 1035 \keys_define:nn { hyp }
allcolors (setup key) 1036 {
    1037     ,allcolors .meta:n =
    1038     {
    1039         ,urlcolor=#1
    1040         ,linkcolor=#1
    1041         ,runcolor=#1
    1042         ,filecolor=#1
    1043         ,menucolor=#1
    1044     }
    1045     ,allcolors .value_required:n = true
    1046 }
1047 }

```

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)

1048 \keys_define:nn { hyp }
1049 {
1050     ,bordercolormodel .choices:nn =
1051     {rgb,cmyk}
1052     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1053     ,bordercolormodel .initial:n ={rgb}
1054 }

1055 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1056 {
1057     \keys_define:nn { hyp }
1058     {
1059         #1bordercolor .code:n =

```

```

1060         {
1061             \tl_if_empty:nTF { ##1 }
1062             {
1063                 \pdfannot_dict_remove:nn
1064                 {link/#2}
1065                 { C }
1066             }
1067             {
1068                 \__hyp_color_set:ne {hyp/color/#1border}{##1}
1069                 \color_export:nVN
1070                 {hyp/color/#1border}
1071                 \g__hyp_bordercolormodel_str
1072                 \l__hyp_tmpa_tl
1073                 \pdfannot_dict_put:nne
1074                 {link/#2}
1075                 { C }
1076                 { [\l__hyp_tmpa_tl] }
1077             }
1078         }
1079     }
1080 }
1081
1082 \keys_define:nn { hyp }
1083 {
1084     ,allbordercolors .meta:n =
1085     {
1086         ,linkbordercolor=#1
1087         ,urlbordercolor =#1
1088         ,filebordercolor=#1
1089         ,menubordercolor=#1
1090         ,runbordercolor =#1
1091     }
1092     ,allbordercolors .value_required:n = true
1093 }
1094

```

11.3.2 Borderwidth and -arc

```

1095 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1096 {
1097     \keys_define:nn { hyp }
1098     {
1099         #1border .code:n =
1100         {
1101             \tl_if_empty:nTF { ##1 }
1102             {
1103                 \pdfannot_dict_remove:nn
1104                 {link/#2}
1105                 { Border }
1106             }
1107             {
1108                 \pdfannot_dict_put:nnn
1109                 {link/#2}
1110                 { Border }

```

```

1111         { [#1] }
1112     }
1113 }
1114 }
1115 }
1116 \keys_define:nn { hyp }
1117 {
1118     ,pdfborder .code:n =
1119     {
1120         \tl_if_empty:nTF { #1 }
1121         {
1122             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1123             {
1124                 \pdfannot_dict_remove:nn
1125                 {link/##2}
1126                 { Border }
1127             }
1128         }
1129         {
1130             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1131             {
1132                 \pdfannot_dict_put:nnn
1133                 {link/##2}
1134                 { Border }
1135                 { [#1] }
1136             }
1137         }
1138     }
1139     ,pdfborder .initial:n = {0~0~1},
1140 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1141 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1142 {
runborderstyle (setup key) 1143     \keys_define:nn { hyp }
fileborderstyle (setup key) 1144     {
menuborderstyle (setup key) 1145         #1borderstyle .code:n =
1146         {
1147             \tl_if_empty:nTF { ##1 }
1148             {
1149                 \pdfannot_dict_remove:nn
1150                 {link/#2}
1151                 { BS }
1152             }
1153             {
1154                 \pdfannot_dict_put:nnn
1155                 {link/#2}
1156                 { BS }
1157                 { <<##1>> }
1158             }
1159         }

```

```

1160     }
1161   }
1162   \keys_define:nn { hyp }
1163   {
1164     ,pdfborderstyle .code:n =
1165     {
1166       \tl_if_empty:nTF { #1 }
1167       {
1168         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1169         {
1170           \pdfannot_dict_remove:nn
1171             {link/##2}
1172             { BS }
1173         }
1174       }
1175       {
1176         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1177         {
1178           \pdfannot_dict_put:nnn
1179             {link/##2}
1180             { BS }
1181             { <<#1>> }
1182         }
1183       }
1184     }
1185     ,pdfborderstyle .initial:n = {},
1186   }

```

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.

```

1187 \cs_new_protected:Npn \__hyp_ocg_init:
1188 {
1189   \pdf_object_new:n { __hyp/OCG/View }
1190   \pdf_object_new:n { __hyp/OCG/Print }
1191   \pdf_object_new:n { __hyp/OCG/config }
1192   \pdf_object_new:n { __hyp/OCG/refarray }
1193   \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1194   {
1195     \pdf_object_ref:n { __hyp/OCG/View }
1196     \c_space_tl
1197     \pdf_object_ref:n { __hyp/OCG/Print }
1198   }
1199   \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1200   {
1201     /Type/OCG
1202     /Name(View)
1203     /Usage
1204     <<
1205     /Print <</PrintState/OFF>>~

```

```

1206         /View <</ViewState/ON >>~
1207     >>
1208 }
1209 \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1210 {
1211     /Type/OCG
1212     /Name(Print)
1213     /Usage
1214     <<
1215         /Print <</PrintState/ON>>~
1216         /View <</ViewState/OFF>>~
1217     >>
1218 }
1219 \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC
1220 \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC
1221 \pdf_object_write:nne { __hyp/OCG/config } { dict }
1222 {
1223     /OFF[\pdf_object_ref:n { __hyp/OCG/Print }]
1224     /AS[
1225         <<
1226             /Event/View
1227             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1228             /Category[/View]
1229         >>
1230         <<
1231             /Event/Print
1232             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1233             /Category[/Print]
1234         >>
1235         <<
1236             /Event/Export
1237             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1238             /Category[/Print]
1239         >>
1240     ]
1241 }
1242 \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1243 \cs_gset:Npn \__hyp_ocg_init: {}
1244 }

```

(End of definition for __hyp_ocg_init:.)

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

```

1245 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1246 {
1247     \hook_gput_code:nnn
1248     {pdfannot/link/#2/begin}
1249     {hyp/ocg}
1250     {
1251         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1252         {
1253             \__hyp_ocg_init:
1254             \group_begin:
1255             \hbox_set:Nw \l__hyp_tmpa_box

```

```

1256     }
1257   }
1258   \hook_gput_code:nnn
1259   {pdfannot/link/#2/end}
1260   {hyp/ocg}
1261   {
1262     \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1263     {
1264       \hbox_set_end:
1265       \mbox
1266       {
1267         \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1268         \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1269         \pdf_emc:
1270         \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1271         \group_begin:
1272         \color_select:n { hyp/color/#1 }
1273         \box_use_drop:N \l__hyp_tmpa_box
1274         \group_end:
1275         \pdf_emc:
1276       }
1277       \group_end:
1278     }
1279   }
1280 }

```

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

```

ocgcolorlink (setup key) 1281 \bool_lazy_or:nnTF
  ocgcolorurl (setup key) 1282 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1283 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1284 {
  ocgcolorrun (setup key) 1285   \keys_define:nn { hyp }
1286   {
1287     ,_ocgcolorlinks .meta:n =
1288     {
1289       ocgcolorlink=#1,
1290       ocgcolorurl=#1,
1291       ocgcolorfile=#1,
1292       ocgcolorrun=#1,
1293       ocgcolormenu=#1
1294     }
1295     ,_ocgcolorlinks .default:n = true
1296   }
1297 }
1298 {
1299   \keys_define:nn { hyp }
1300   {
1301     ,_ocgcolorlinks .code:n =
1302     {
1303       \msg_warning:nnee
1304       { hyp }
1305       { ignore-deprecated-or-unknown-option-in-pdf-version }
1306       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1307     }

```

```

1308     }
1309 }
1310
1311 \keys_define:nn { hyp }
1312 {
1313   ,ocgcolorlinks .choice:
1314   ,ocgcolorlinks / true .meta:n =
1315   {
1316     pdfborder      = {0~0~0},
1317     pdfborderstyle = {},
1318     colorlinks     = false,
1319     _ocgcolorlinks = true
1320   }
1321   ,ocgcolorlinks / false .meta:n =
1322   {
1323     _ocgcolorlinks = false
1324   }
1325   ,ocgcolorlinks .default:n = {true}
1326 }
1327
1328 \seq_map_inline:Nn \c__hyp_annot_types_seq
1329 {
1330   \bool_lazy_or:nnTF
1331   { \pdf_version_compare_p:Nn > {1.4} }
1332   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1333   {
1334     \keys_define:nn { hyp }
1335     {
1336       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1337     }
1338   }
1339   {
1340     \keys_define:nn { hyp }
1341     {
1342       ,ocgcolor#1 .code:n=
1343       {
1344         \msg_warning:nnee
1345         { hyp }
1346         { ignore-deprecated-or-unknown-option-in-pdf-version }
1347         { ocgcolor#1 }
1348         { \pdf_version_major:.\pdf_version_minor: }
1349       }
1350     }
1351   }
1352 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1353 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1354 {
1355   \keys_define:nn { hyp }
1356   {
1357     ,#1highlight .choices:nn =

```



```

1358     { /I, /N, /O, /P}
1359     {
1360         \pdfannot_dict_put:nnn
1361         {link/#2}
1362         { H }
1363         { ##1 }
1364     }
1365     ,#1highlight / .code:n =
1366     {
1367         \pdfannot_dict_remove:nn
1368         {link/#2}
1369         { H }
1370     }
1371     }
1372     ,#1highlight / unknown .code:n =
1373     {
1374         \msg_warning:nneee { hyp } { unknown-choice+empty }
1375         { pdfhighlight }
1376         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1377         { \exp_not:n {##1} }
1378     }
1379 }
1380 }
1381 }
1382
1383 \keys_define:nn { hyp }
1384 {
1385     ,pdfhighlight .choices:nn =
1386     { /I, /N, /O, /P}
1387     {
1388         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1389         {
1390             \pdfannot_dict_put:nnn
1391             {link/####2}
1392             { H }
1393             { #1 }
1394         }
1395     }
1396     ,pdfhighlight / .code:n =
1397     {
1398         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1399         {
1400             \pdfannot_dict_remove:nn
1401             {link/##2}
1402             { H }
1403         }
1404     }
1405     ,pdfhighlight .initial:n = {/I},
1406     ,pdfhighlight / unknown .code:n =
1407     {
1408         \msg_warning:nneee { hyp } { unknown-choice+empty }
1409         { pdfhighlight }
1410         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1411     }

```

```

1412         { \exp_not:n {#1} }
1413     }
1414 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1415 \keys_define:nn { hyp }
hideurl (setup key) 1416 {
hidefile (setup key) 1417     hidelinks .meta:n =
hiderun (setup key) 1418     {
hidemenu (setup key) 1419         ,colorlinks      = false
1420         ,ocgcolorlinks = false
1421         ,pdfborder      = { 0~0~0 }
1422         ,pdfborderstyle=
1423     }
1424 }
1425
1426 \seq_map_inline:Nn \c__hyp_annot_types_seq
1427 {
1428     \keys_define:nn { hyp }
1429     {
1430         hide#1 .meta:n =
1431         {
1432             ,color#1      = false
1433             ,ocgcolor#1   = false
1434             ,#1border     = { 0~0~0 }
1435             ,#1borderstyle =
1436         }
1437     }
1438 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1439 \keys_define:nn { hyp }
1440 {
1441     colorscheme .code:n =
1442     {
1443         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1444         {
1445             \keys_set:nn { hyp }
1446             {
1447                 ##1 = ##2
1448             }
1449         }
1450     }
1451 }
1452 \keys_set:nn { hyp } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```
unicode (setup key)
pdfencoding (setup key) 1453 \keys_define:nn { hyp }
pdfversion (setup key) 1454 {
1455     ,unicode .code:n = {}
1456     ,pdfencoding .code:n = {}
1457     ,pdfversion .code:n =
1458     {
1459         \msg_warning:nn { hyp }{ pdfversion-disabled }
1460     }
1461 }
1462 %
```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```
verbose (setup key)
debug (setup key) 1463 \keys_define:nn { hyp }
draft (setup key) 1464 {
1465     ,verbose .legacy_if_set:n = {Hy@verbose}
1466     ,debug .legacy_if_set:n = {Hy@verbose}
1467 }
1468 \keys_define:nn { hyp }
1469 {
1470     ,draft .code:n =
1471     {
1472         \Hy@drafttrue
1473         \PassOptionsToPackage{draft}{bookmark}
1474     }
1475     ,final .code:n =
1476     {
1477         \Hy@finaltrue
1478         \PassOptionsToPackage{final}{bookmark}
1479     }
1480 }

extension (setup key)
hypertextnames (setup key) 1481 \keys_define:nn { hyp }
naturalnames (setup key) 1482 {
1483     ,extension .tl_set:N = \XR@ext
1484     ,extension .initial:n= pdf
1485     ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
1486     ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
1487     ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
1488     ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1489     ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1490     ,plainpages .legacy_if_set:n = {Hy@plainpages}
1491 }
```

```

1492
1493 \keys_define:nn { hyp }
1494 {
1495   ,linktoc .choices:nn = { none, section, all, page }
1496   {
1497     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1498   }
1499   ,linktoc / unknown .code:n =
1500   {
1501     \msg_warning:nneee { hyp } { unknown-choice }
1502     { linktoc }
1503     { none, section, all, page }
1504     { \exp_not:n {#1} }
1505   }
1506   ,linktocpage .choice:
1507   ,linktocpage / true .meta:n = {linktoc=page}
1508   ,linktocpage / false .meta:n = {linktoc=section}
1509   ,linktocpage .default:n = true
1510 }
1511
link (setup key) This booleans allow to disable the link types.
url (setup key) 1512 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1513 {
menu (setup key) 1514   \keys_define:nn { hyp }
run (setup key) 1515   {
1516     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1517   }
1518 }

1519 \keys_define:nn { hyp }
1520 {
1521   ,baseurl .code:n =
1522   {
1523     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1524     \tl_if_empty:NTF \l__hyp_tmpa_tl
1525     {
1526       \pdfmanagement_remove:nn {Catalog} { URI }
1527     }
1528     {
1529       \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1530     }
1531     \__hyp_store_metadata:nn {baseurl}{#1}
1532   }
1533   %only false does something ...
1534   ,bookmarks .choice:
1535   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1536   ,bookmarks / true .code:n = {}
1537   ,bookmarks .default:n = {true}
1538   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1539   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1540   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1541   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1542   ,pdfcenterwindow .choice:

```

```

1543 ,pdfcenterwindow / false .code:n =
1544 {
1545     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1546 }
1547 ,pdfcenterwindow / true .code:n =
1548 {
1549     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1550 }
1551 ,pdfcenterwindow / .code:n =
1552 {
1553     \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1554 }
1555 ,pdfcenterwindow / unknown .code:n =
1556 {
1557     \msg_warning:nnee { hyp } { no-bool }
1558     { pdfcenterwindow }
1559     { \exp_not:n {#1} }
1560 }
1561 ,pdfcenterwindow .default:n = true
1562 ,pdfdirection .choice:
1563 ,pdfdirection / L2R .code:n =
1564 {
1565     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1566 }
1567 ,pdfdirection / R2L .code:n =
1568 {
1569     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1570 }
1571 ,pdfdirection / .code:n =
1572 {
1573     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1574 }
1575 ,pdfdirection / unknown .code:n =
1576 {
1577     \msg_warning:nneee { hyp } { unknown-choice+empty }
1578     { pdfdirection }
1579     { L2R , R2L }
1580     { \exp_not:n {#1} }
1581 }
1582 ,pdfdisplaydoctitle .choice:
1583 ,pdfdisplaydoctitle / false .code:n =
1584 {
1585     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1586 }
1587 ,pdfdisplaydoctitle / true .code:n =
1588 {
1589     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1590 }
1591 ,pdfdisplaydoctitle .default:n = true
1592 ,pdfduplex .choices:nn =
1593 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1594 {
1595     \pdf_version_compare:NnTF > {1.6}
1596     {

```

```

1597         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1598         { PrintDuplex } { /#1 }
1599     }
1600     {
1601         \msg_warning:nnee
1602         {hyp}
1603         {ignore-deprecated-or-unknown-option-in-pdf-version}
1604         {pdfduplex}
1605         {\pdf_version:}
1606     }
1607 }%
1608 ,pdfduplex / .code:n =
1609 {
1610     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1611 }
1612 ,pdfduplex / unknown .code:n =
1613 {
1614     \msg_warning:nneee { hyp } { unknown-choice+empty }
1615     { pdfduplex }
1616     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1617     { \exp_not:n {#1} }
1618 }
1619 ,pdffitwindow .choice:
1620 ,pdffitwindow / false .code:n =
1621 {
1622     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1623 }
1624 ,pdffitwindow / true .code:n =
1625 {
1626     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1627 }
1628 ,pdffitwindow / .code:n =
1629 {
1630     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1631 }
1632 ,pdffitwindow .default:n = true
1633 ,pdffitwindow / unknown .code:n =
1634 {
1635     \msg_warning:nnee { hyp } { no-bool }
1636     { pdffitwindow }
1637     { \exp_not:n {#1} }
1638 }
1639 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1640 ,pdflinkmargin .initial:n = {1pt}
1641 ,pdfmenubar .choice:
1642 ,pdfmenubar / true .code:n =
1643 {
1644     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1645 }
1646 ,pdfmenubar / false .code:n =
1647 {
1648     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1649     { HideMenubar } { true }
1650 }

```

```

1651 ,pdfmenubar / .code:n =
1652 {
1653     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1654 }
1655 ,pdfmenubar .default:n = true
1656 ,pdfmenubar / unknown .code:n =
1657 {
1658     \msg_warning:nnee { hyp } { no-bool }
1659     { pdfmenubar }
1660     { \exp_not:n {#1} }
1661 }
1662 ,pdfnewwindow .choice:
1663 ,pdfnewwindow / true .code:n =
1664 {
1665     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1666     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1667 }
1668 ,pdfnewwindow / false .code:n =
1669 {
1670     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1671     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1672 }
1673 ,pdfnewwindow / .code:n =
1674 {
1675     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1676     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1677 }
1678 ,pdfnonfullscreenpagemode .choices:nn =
1679 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1680 {
1681     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1682     { NonFullScreenPageMode } {/#1}
1683 }
1684 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1685 {
1686     \pdf_version_compare:NnTF < {1.6}
1687     {
1688         %message
1689     }
1690     {
1691         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1692         {NonFullScreenPageMode}{/UseAttachments}
1693     }
1694 }
1695 ,pdfnonfullscreenpagemode / .code:n =
1696 {
1697     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1698 }
1699 ,pdfnonfullscreenpagemode / unknown .code:n =
1700 {
1701     \msg_warning:nneee { hyp } { unknown-choice+empty }
1702     { pdfnonfullscreenpagemode }
1703     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1704     { \exp_not:n {#1} }

```

```

1705     }
1706 ,pdfnumcopies .code:n =
1707 {
1708     \pdf_version_compare:NnTF > {1.6}
1709     {
1710         \tl_if_empty:nTF {#1}
1711         {
1712             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1713         }
1714         {
1715             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1716             {NumCopies}{#1}
1717         }
1718     }
1719     {
1720         \msg_warning:nnee
1721         {hyp}
1722         {ignore-deprecated-or-unknown-option-in-pdf-version}
1723         {pdfnumcopies}
1724         {\pdf_version:}
1725     }
1726 }
1727 ,pdfpagelayout .choices:nn =
1728 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1729 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1730 ,pdfpagelayout / .code:n =
1731 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1732 ,pdfpagelayout / unknown .code:n =
1733 {
1734     \msg_warning:nneee { hyp } { unknown-choice+empty }
1735     { pdfpagelayout }
1736     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1737     { \exp_not:n {#1} }
1738 }
1739 ,pdfpagemode .choices:nn =
1740 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1741 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1742 ,pdfpagemode / UseAttachments .code:n =
1743 {
1744     \pdf_version_compare:NnTF > {1.5}
1745     {
1746         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1747     }
1748     {
1749         \msg_warning:nnee
1750         {hyp}
1751         {ignore-deprecated-or-unknown-value-in-pdf-version}
1752         {UseAttachments}
1753         {\pdf_version:}
1754     }
1755 }
1756 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1757 ,pdfpagemode / unknown .code:n =
1758 {

```



```

1759     \msg_warning:nneee { hyp } { unknown-choice+empty }
1760     { pdfpagemode }
1761     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1762     { \exp_not:n {#1} }
1763 }
1764 ,pdfpagescrop .code:n =
1765 {
1766     \tl_if_empty:nTF {#1} %or blank?
1767     {
1768         \pdfmanagement_remove:nn {Pages} { CropBox }
1769     }
1770     {
1771         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1772     }
1773 }
1774 ,pdfpicktraybypdfsize .choice:
1775 ,pdfpicktraybypdfsize / true .code:n =
1776 {
1777     \pdf_version_compare:NnTF > {1.6}
1778     {
1779         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1780         { PickTrayByPDFSize } { true }
1781     }
1782     {
1783         \msg_warning:nnee
1784         {hyp}
1785         {ignore-deprecated-or-unknown-option-in-pdf-version}
1786         {pdfpicktraybypdfsize}
1787         {\pdf_version:}
1788     }
1789 }
1790 ,pdfpicktraybypdfsize / false .code:n =
1791 {
1792     \pdf_version_compare:NnTF > {1.6}
1793     {
1794         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1795         { PickTrayByPDFSize } { false }
1796     }
1797     {
1798         \msg_warning:nnee
1799         {hyp}
1800         {ignore-deprecated-or-unknown-option-in-pdf-version}
1801         {pdfpicktraybypdfsize}
1802         {\pdf_version:}
1803     }
1804 }
1805 ,pdfpicktraybypdfsize / .code:n =
1806 {
1807     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1808 }
1809 ,pdfpicktraybypdfsize / unknown .code:n =
1810 {
1811     \msg_warning:nnee { hyp } { no-bool }
1812     { picktraybypdfsize }

```

```

1813         { \exp_not:n {#1} }
1814     }
1815 ,pdfprintarea .choices:nn =
1816 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1817 {
1818     \pdf_version_compare:NnTF < {2.0}
1819     {
1820         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1821         { PrintArea } { /#1 }
1822     }
1823     {
1824         \msg_warning:nnee
1825         {hyp}
1826         {ignore-deprecated-or-unknown-option-in-pdf-version}
1827         {pdfprintarea}
1828         {\pdf_version:}
1829     }
1830 }%
1831 ,pdfprintarea / .code:n =
1832 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1833 ,pdfprintarea / unknown .code:n =
1834 {
1835     \msg_warning:nnee { hyp } { unknown-choice+empty }
1836     { pdfprintarea }
1837     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1838     { \exp_not:n {#1} }
1839 }
1840 ,pdfprintclip .choices:nn =
1841 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1842 {
1843     \pdf_version_compare:NnTF < {2.0}
1844     {
1845         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1846         { PrintClip } { /#1 }
1847     }
1848     {
1849         \msg_warning:nnee
1850         {hyp}
1851         {ignore-deprecated-or-unknown-option-in-pdf-version}
1852         {pdfprintclip}
1853         {\pdf_version:}
1854     }
1855 }%
1856 ,pdfprintclip / .code:n =
1857 {
1858     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1859 }
1860 ,pdfprintclip / unknown .code:n =
1861 {
1862     \msg_warning:nnee
1863     { hyp }
1864     { unknown-choice+empty }
1865     { pdfprintclip }
1866     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1867         { \exp_not:n {#1} }
1868     }
1869 ,pdfprintpagerange .code:n =
1870 {
1871     \pdf_version_compare:NnTF > {1.6}
1872     {
1873         \tl_if_empty:nTF { #1}
1874         {
1875             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1876             { PrintPageRange }
1877         }
1878         {
1879             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1880             {PrintPageRange}{[#1]}
1881         }
1882     }
1883     {
1884         \msg_warning:nnee
1885         {hyp}
1886         {ignore-deprecated-or-unknown-option-in-pdf-version}
1887         {pdfprintpagerange}
1888         {\pdf_version:}
1889     }
1890 }
1891 ,pdfprintscaling .choices:nn =
1892 { None, AppDefault }
1893 {
1894     \pdf_version_compare:NnTF > {1.5}
1895     {
1896         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1897         { PrintScaling } { /#1 }
1898     }
1899     {
1900         \msg_warning:nnee
1901         {hyp}
1902         {ignore-deprecated-or-unknown-option-in-pdf-version}
1903         {pdfprintscaling}
1904         {\pdf_version:}
1905     }
1906 }%
1907 ,pdfprintscaling / .code:n =
1908 {
1909     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1910 }
1911 ,pdfprintscaling / unknown .code:n =
1912 {
1913     \msg_warning:nneee { hyp } { unknown-choice+empty }
1914     { pdfprintarea }
1915     { None, AppDefault }
1916     { \exp_not:n {#1} }
1917 }
1918 ,pdfremotestartview .code:n =
1919 {
1920     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}

```

```

1921     \exp_args:NNV
1922     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1923     {
1924         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1925     }
1926     {
1927         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1928         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1929     }
1930 }
1931 ,pdfremotestartview .initial:n = {Fit}
1932 % pdfstartpage is special as it shares code with pdfstartview
1933 ,pdfstartpage .code:n =
1934 {
1935     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1936     \bool_if:nTF
1937     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1938     {
1939         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1940     }
1941     {
1942         \pdfmanagement_add:nne {Catalog} { OpenAction }
1943         {
1944             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1945         }
1946     }
1947 }
1948 ,pdfstartpage .initial:n =1
1949 ,pdfstartview .code:n =
1950 {
1951     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1952     \exp_args:NNV
1953     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1954     {
1955         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1956     }
1957     {
1958         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1959         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1960     }
1961     \bool_if:nTF
1962     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1963     {
1964         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1965     }
1966     {
1967         \pdfmanagement_add:nne {Catalog} { OpenAction }
1968         {
1969             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1970         }
1971     }
1972 }
1973 ,pdfstartview .initial:n = Fit
1974 ,pdftoolbar .choice:

```

```

1975 ,pdftoolbar / true .code:n =
1976 {
1977     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1978 }
1979 ,pdftoolbar / false .code:n =
1980 {
1981     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1982     { HideToolbar } { true }
1983 }
1984 ,pdftoolbar / true .code:n =
1985 {
1986     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
1987 }
1988 ,pdftoolbar .default:n = true
1989 ,pdftoolbar / unknown .code:n =
1990 {
1991     \msg_warning:nnee { hyp } { no-bool }
1992     { pdftoolbar }
1993     { \exp_not:n {#1} }
1994 }
1995 % pdfview see below.
1996 ,pdfviewarea .choices:nn =
1997 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1998 {
1999     \pdf_version_compare:NnTF < {2.0}
2000     {
2001         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2002         { ViewArea } { /#1 }
2003     }
2004     {
2005         \msg_warning:nnee
2006         {hyp}
2007         {ignore-deprecated-or-unknown-option-in-pdf-version}
2008         {pdfviewarea}
2009         {\pdf_version:}
2010     }
2011 }%
2012 ,pdfviewarea / .code:n =
2013 {
2014     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2015 }
2016 ,pdfviewarea / unknown .code:n =
2017 {
2018     \msg_warning:nneee { hyp } { unknown-choice+empty }
2019     { pdfviewarea }
2020     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2021     { \exp_not:n {#1} }
2022 }
2023 ,pdfviewclip .choices:nn =
2024 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2025 {
2026     \pdf_version_compare:NnTF < {2.0}
2027     {
2028         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }

```

```

2029         { ViewClip } { /#1 }
2030     }
2031     {
2032         \msg_warning:nnee
2033         {hyp}
2034         {ignore-deprecated-or-unknown-option-in-pdf-version}
2035         {pdfviewclip}
2036         {\pdf_version:}
2037     }
2038 }%
2039 ,pdfviewclip / .code:n =
2040 {
2041     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2042 }
2043 ,pdfviewclip / unknown .code:n =
2044 {
2045     \msg_warning:nneee { hyp } { unknown-choice+empty }
2046     { pdfviewclip }
2047     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2048     { \exp_not:n {#1} }
2049 }
2050 ,pdfwindowui .choice:
2051 ,pdfwindowui / true .code:n =
2052 {
2053     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2054 }
2055 ,pdfwindowui / false .code:n =
2056 {
2057     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2058     { HideWindowUI } { true }
2059 }
2060 ,pdfwindowui / .code:n =
2061 {
2062     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2063 }
2064 ,pdfwindowui / unknown .code:n =
2065 {
2066     \msg_warning:nnee { hyp } { no-bool }
2067     { pdfwindowui }
2068     { \exp_not:n {#1} }
2069 }
2070 ,pdfwindowui .default:n = true
2071 }

```

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

```

2072 \keys_define:nn { hyp }
2073 {
2074     ,pdfview .code:n =
2075     {
2076         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2077         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2078         {
2079             { xyz }
2080             {

```

```

2081 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2082 {
2083   \seq_get_right:Nn \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2084   \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2085   {
2086     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2087   }
2088   {
2089     \tl_set:Ne \l__hyp_dest_pdfview_tl
2090     {
2091       \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2092     }
2093   }
2094 }
2095 {
2096   \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2097 }
2098 }
2099 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2100 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2101 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2102 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2103 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2104 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2105 { fitr }
2106 {
2107   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2108   {
2109     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2110   }
2111   {
2112     %ensure 4 values ...
2113     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2114     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2115     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2116     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2117     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2118     {
2119       \fp_eval:n
2120       {
2121         round
2122         (
2123           abs
2124           (
2125             \seq_item:Nn\l__hyp_tmpa_seq{4}
2126             -
2127             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2128           ),
2129           3
2130         )
2131       }bp
2132     }{}
2133     \box_set_dp:Nn \l__hyp_dest_box
2134     {

```

```

2135         \fp_eval:n
2136         {
2137             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2138         }bp
2139     }
2140     \box_set_ht:Nn \l__hyp_dest_box
2141     {
2142         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2143     }
2144 }
2145 }
2146 }
2147 {
2148     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2149     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2150 }
2151 }
2152 ,pdfview .initial:n = {xyz}
2153 }

```

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.

```

2154 \keys_define:nn { hyp }
2155 {
2156     ,pdflang .code:n =
2157     {
2158         \tl_if_empty:nF { #1 }
2159         {
2160             \pdfmanagement_add:nne {Catalog} { Lang } { { (#1) }
2161             \AddToDocumentProperties[document]{lang}{#1}
2162         }
2163     }
2164 }

```

12.3.1 “info keys”

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

pdftitle (*setup key*) Creator and Producer can’t be removed with the pdfmanagement, but we allow to set an

pdfcreator (*setup key*) empty value. If the value begin with an optional argument, we assume a multilanguage

pdfsubject (*setup key*) clist and use only the first value.

```

pdfproducer (setup key) 2165 \regex_new:N\l__hyp_optlang_regex
pdfkeywords (setup key) 2166 \regex_set:Nn\l__hyp_optlang_regex {\A\([([A-Za-z\-\-]+)\)(.*)}
2167 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2168 {
2169     \keys_define:nn { hyp }
2170     {
2171         pdf#1 .code:n =
2172         {
2173             \tl_if_blank:nTF {##1}

```



```

2174     {
2175         \str_case:nnF { #1 }
2176         {
2177             {creator}
2178             {
2179                 \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2180                 \pdfmanagement_add:nne {Info}{Creator}{()}
2181             }
2182             {producer}
2183             {
2184                 \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2185                 \pdfmanagement_add:nne {Info}{Producer}{()}
2186             }
2187         }
2188         {
2189             \pdfmanagement_remove:nn {Info}{#2}
2190         }
2191     }
2192     {
2193         \tl_set:Ne\l__hyp_tmpa_tl {\clist_item:nn{##1}{1}}
2194         \exp_args:NNV
2195         \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpa_tl\l__hyp_tmpa_s
2196         \seq_if_empty:NTF\l__hyp_tmpa_seq
2197         {
2198             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2199         }
2200         {
2201             \exp_args:Ne
2202             \__hyp_text_pdfstring_info:nN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2203         }
2204         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2205         {
2206             \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2207         }
2208     }
2209     \__hyp_store_metadata:nn {pdf#1}{##1}
2210 }
2211 }
2212 \keys_define:nn { hyp / info }
2213 {
2214     #2 .code:n =
2215     {
2216         \tl_if_blank:nTF {##1}
2217         {
2218             \pdfmanagement_remove:nn {Info}{#2}
2219         }
2220         {
2221             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2222             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2223             {
2224                 \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2225             }
2226         }
2227     }
2228     \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}

```

```

2228     }
2229     ,unknown .code:n =
2230     {
2231         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2232         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2233         {
2234             \exp_args:Nno
2235             \pdfmanagement_add:nne {Info}
2236             { \l_keys_key_str } {\l__hyp_tmpa_str}
2237         }
2238     }
2239 }
2240 }
2241 \__hyp_setup_info_key:nn {author} {Author}
2242 \__hyp_setup_info_key:nn {title} {Title}
2243 \__hyp_setup_info_key:nn {producer} {Producer}
2244 \__hyp_setup_info_key:nn {creator} {Creator}
2245 % ignored key: addtopdfcreator
2246 \__hyp_setup_info_key:nn {subject} {Subject}
2247 \__hyp_setup_info_key:nn {keywords} {Keywords}

```

pdfcreationdate (setup key) These keys are not really needed. We store them too in the container. CreationDate and
pdfmoddate (setup key) ModDate should not use the hex encoding.

```

pdfmetadate (setup key)
2248 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2249 {
2250     \keys_define:nn { hyp }
2251     {
2252         pdf#1 .code:n =
2253         {
2254             \tl_if_blank:nTF {##1}
2255             {
2256                 \pdfmanagement_remove:nn {Info}{#2}
2257             }
2258             {
2259                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2260             }
2261             \__hyp_store_metadata:nn {pdf#1}{##1}
2262             \AddToDocumentProperties[document]{#1}{##1}
2263         }
2264     }
2265     \keys_define:nn { hyp / info }
2266     {
2267         #2 .code:n =
2268         {
2269             \tl_if_blank:nTF {##1}
2270             {
2271                 \pdfmanagement_remove:nn {Info}{#2}
2272             }
2273             {
2274                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2275             }
2276             \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2277         }
2278     }

```

```

2279 }
2280
2281 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2282 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2283 \keys_define:nn { hyp }
2284 {
2285     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2286 }

```

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

```

2287 \keys_define:nn { hyp }
2288 {
2289     ,pdftrapped .code:n =
2290     {
2291         \exp_args:Nne
2292         \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2293     }
2294     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2295     {
2296         \pdfmanagement_add:nne {Info}{Trapped}
2297         {/
2298             \str_uppercase:f { \str_head:n { #1 } }
2299             \str_lowercase:f { \str_tail:n { #1 } }
2300         }
2301         \__hyp_store_metadata:ne {pdftrapped}
2302         {
2303             \str_uppercase:f { \str_head:n { #1 } }
2304             \str_lowercase:f { \str_tail:n { #1 } }
2305         }
2306     }
2307     ,_pdftrapped / unknown .code:n =
2308     {
2309         \msg_warning:nneee { hyp } { unknown-choice }
2310         { pdftrapped }
2311         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2312         { \exp_not:n {#1} }
2313     }
2314 }

```

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

```

2315 \keys_define:nn { hyp }
2316 {
2317     pdfinfo .code:n =
2318     {
2319         \keys_set:nn { hyp / info } { #1 }
2320     }
2321 }

```

Now we set some default values

```

2322 \keys_set:nn { hyp } {pdfcreator = LaTeX-with-hyperref}
2323 \keys_set:nn { hyp } {pdfauthor = }
2324 \keys_set:nn { hyp } {pdftitle = }
2325 \keys_set:nn { hyp } {pdfsubject = }

```

12.4 hyperxmp keys

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

```
2326 \clist_map_inline:nn
2327 {
2328   ,pdfcopyright
2329   ,pdftype
2330   ,pdflicenseurl
2331   ,pdfauthortitle
2332   ,pdfcaptionwriter
2333   ,pdfmetalang
2334   ,pdfsource
2335   ,pdfdocumentid
2336   ,pdfinstanceid
2337   ,pdfversionid
2338   ,pdfrendition
2339   ,pdfpublication
2340   ,pdfpubtype
2341   ,pdfbytes
2342   ,pdfnumpages
2343   ,pdfissn
2344   ,pdfeissn
2345   ,pdfisbn
2346   ,pdfbookedition
2347   ,pdfpublisher
2348   ,pdfvolumenum
2349   ,pdfissuenum
2350   ,pdfpagerange
2351   ,pdfdoi
2352   ,pdfurl
2353   ,pdfidentifier
2354   ,pdfsubtitle
2355   ,pdfpubstatus
2356   ,pdfcontactaddress
2357   ,pdfcontactcity
2358   ,pdfcontactregion
2359   ,pdfcontactpostcode
2360   ,pdfcontactcountry
2361   ,pdfcontactphone
2362   ,pdfcontactemail
2363   ,pdfcontacturl
2364   ,pdfdate
2365 }
2366 {
2367   \keys_define:nn { hyp }
2368   {
2369     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2370   }
2371 }
2372
```

12.5 Transitions

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

```

2373 \keys_define:nn { hyp }
2374 {
2375   pdfpageduration .code:n =
2376   {
2377     \tl_if_blank:nTF { #1 }
2378     {
2379       \pdfmanagement_remove:nn {Page}{Dur}
2380     }
2381     {
2382       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2383     }
2384   }
2385 }

```

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

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

The keys allowed in the argument of `pdfpagetransition` are

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

```

2386 \keys_define:nn { hyp }
2387 {
2388   pdfpagetransition .code:n =
2389   {
2390     \tl_if_blank:nTF {#1}
2391     {
2392       \pdfmanagement_remove:nn {Page}{Trans}
2393     }

```

```

2394     {
2395         \group_begin:
2396         \keys_set:nn { hyp / trans }{style=R,#1}
2397         \pdf_object_unnamed_write:ne { dict }
2398         {
2399             \pdfdict_use:n {l__hyp_page/Trans}
2400         }
2401         \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2402         \group_end:
2403     }
2404 }
2405 }
2406 \keys_define:nn { hyp / trans }
2407 {
2408     ,style .choices:nn =
2409     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2410     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2411     ,style / unknown .code:n =
2412     {
2413         \msg_warning:nneee { hyp } { unknown-choice }
2414         { trans / style }
2415         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2416         { \exp_not:n {#1} }
2417     }
2418     ,duration .code:n =
2419     {
2420         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2421     }
2422     ,direction .choices:nn =
2423     {H,V}
2424     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2425     ,direction .choices:nn =
2426     {0,90,180,270,315}
2427     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2428     ,direction / None .code:n =
2429     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2430     ,direction / unknown .code:n =
2431     {
2432         \msg_warning:nneee { hyp } { unknown-choice }
2433         { trans / direction }
2434         {
2435             H~(horizontal,~only~Split,~Blinds),
2436             V~(vertical,~only~Split,~Blinds),
2437             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2438             90~(bottom~to~top,~only~Wipe),
2439             180~(right~to~left,~only~Wipe),
2440             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2441             315~(top~left~to~bottom,~only~Glitter),
2442             None~(only~Fly)
2443         }
2444         { \exp_not:n {#1} }
2445     }
2446     ,motion .choices:nn =
2447     {I,0}

```

```

2448 { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2449 ,motion / unknown .code:n =
2450 {
2451   \msg_warning:nneee { hyp } { unknown-choice }
2452   { trans / motion }
2453   { I~(inwards) , O~(outwards) }
2454   { \exp_not:n {#1} }
2455 }
2456 ,scale .code:n =
2457 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2458 ,opaque .choices:nn = {true,false}
2459 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2460 ,opaque / unknown .code:n =
2461 {
2462   \msg_warning:nneee { hyp } { unknown-choice }
2463   { trans / B }
2464   { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2465   { \exp_not:n {#1} }
2466 }
2467 % try to set unknown keys as style
2468 ,unknown .code:n =
2469 {
2470   % warning ...
2471   \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2472 }
2473 }

```

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

```

2474 \keys_set_known:nv{ hyp }{opt@hyperref.sty}

```

Unfinished Form field code

```

2475 \NewDocumentCommand \MakeFieldObject { m m }
2476 {
2477   \pdfxform_new:nnn { #2 }{ } { #1 }
2478 }
2479
2480
2481 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2482 \prop_new:N \g__hyp_AcroForm_Fields_prop
2483
2484 \let\HyField@afields\@empty
2485 \let\HyField@cofields\@empty
2486 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2487
2488 % the value doesn't matter, but with a prop we avoid duplicates and it is
2489 % clearly faster than removing them from a sequence
2490 \def\HyField@AuxAddToFields#1
2491 {
2492   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2493 }%
2494
2495 %fields with empty key get a value too -- lets hope that
2496 %this give the expected behaviour
2497 \def\HyField@AuxAddToCoFields #1 #2
2498 {

```

```

2499     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2500   }
2501
2502   \Hy@AtBeginDocument
2503   {
2504     \if@filesw
2505       \immediate\write\@mainaux{%
2506         \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2507       }%
2508       \immediate\write\@mainaux{%
2509         \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2510       }%
2511     \fi
2512     \let\HyField@AfterAuxOpen\@firstofone
2513   }%
2514
2515   \def\HyField@AddToFields
2516   {
2517     \exp_args:Nc\HyField__hypAddToFields
2518     {
2519       \pdfannot_box_ref_last:
2520     }
2521     \ifx\Fld@calculate@code\@empty
2522     \else
2523       \begingroup
2524         \Hy@safe@activetrue
2525         \edef\Hy@temp{%
2526           \endgroup
2527           \if@filesw
2528             \write\@mainaux
2529             {
2530               \string\HyField@AuxAddToCoFields
2531               {
2532                 \Fld@calculate@sortkey
2533               }
2534               {
2535                 \pdfannot_box_ref_last:
2536               }
2537             }
2538           \fi
2539         }%
2540         \Hy@temp
2541       \fi
2542     }%
2543
2544   \def\HyField__hypAddToFields#1{
2545     \HyField@AfterAuxOpen{%
2546       \if@filesw
2547         \write\@mainaux{%
2548           \string\HyField@AuxAddToFields{#1}%
2549         }%
2550       \fi
2551     }%
2552   }%

```



```

2553
2554 \ExplSyntaxOff
2555 \ExplSyntaxOn
2556
2557 \def\@Form[#1]
2558 {
2559   \kvsetkeys{Form}{#1}
2560   \pdf@ifdraftmode{}
2561   {
2562     \Hy@FormObjects
2563     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2564     {
2565       \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2566       %\pdfmanagement_show:n { Catalog / AcroForm }
2567     }
2568     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2569     {
2570       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2571       {
2572         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2573       }
2574       \seq_sort:Nn \l__hyp_tmpa_seq
2575       {
2576         \str_compare:nNnTF {##1} > {##2}
2577         { \sort_return_swapped: }
2578         { \sort_return_same: }
2579       }
2580       \seq_map_inline:Nn \l__hyp_tmpa_seq
2581       {
2582         \pdfmanagement_add:nne { Catalog / AcroForm }
2583         { CO }
2584         {
2585           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2586         }
2587       }
2588     }
2589     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2590     {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2591     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2592     {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2593     \pdfmanagement_add:nne {Catalog /AcroForm}
2594     {DA}{(/Helv~10~Tf~0~g)}
2595     \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2596     {
2597       \legacy_if:nT { HyField@NeedAppearances }
2598       {
2599         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2600       }
2601     }
2602     {
2603       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2604     }
2605   }
2606 }

```

```

2607 \ExplSyntaxOff
2608 \let\@endForm\@empty
2609 \let\HyAnn@AbsPageLabel\@empty
2610 \let\Fld@pageobjref\@empty
2611
2612 \ExplSyntaxOn
2613 \newcount\HyAnn@Count
2614 \HyAnn@Count=\z@
2615 \def\HyAnn@AbsPageLabel
2616 {
2617   \global\advance\HyAnn@Count by\@ne
2618   %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2619   %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2620   %\zref@refused{HyAnn@\the\HyAnn@Count}%
2621   \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2622   \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2623 }%
2624 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2625 \def\Fld@pageobjref
2626 {
2627   \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2628   {
2629     /P~\pdf_pageobject_ref:e
2630     {
2631       \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2632     }
2633   }
2634 }
2635 \ExplSyntaxOff
2636 \ExplSyntaxOn
2637 %% check if the attr should be set through
2638 %% hooks.
2639 %% check if options are missing.
2640 \def\@TextField[#1]#2{% parameters, label
2641   \def\Fld@name{#2}%
2642   \let\Fld@default\@empty
2643   \let\Fld@value\@empty
2644   \def\Fld@width{\DefaultWidthofText}%
2645   \def\Fld@height{%
2646     \ifFld@multiline
2647       \DefaultHeightofTextMultiline
2648     \else
2649       \DefaultHeightofText
2650     \fi
2651   }%
2652   \begingroup
2653     \expandafter\HyField@SetKeys\expandafter{%
2654       \DefaultOptionsofText,#1%
2655     }%
2656     \PDFForm@Name
2657     \HyField@FlagsText
2658     \ifFld@hidden\def\Fld@width{1sp}\fi
2659     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2660     \LayoutTextField{#2}{%

```

```

2661 \leavevmode
2662 \HyAnn@AbsPageLabel
2663 \Hy@escapeform\PDFForm@Text
2664 \pdfannot_box:nnnn
2665 {\Fld@width}
2666 {\Fld@height}
2667 {Opt} %is this correct?
2668 {\PDFForm@Text}
2669 \MakeTextField{\Fld@width}{\Fld@height}
2670 \HyField@AddToFields
2671 }%
2672 \endgroup
2673 }
2674 \providecommand\@curropt{}
2675 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2676 \def\Fld@name{#2}
2677 \let\Fld@default\relax
2678 \let\Fld@value\relax
2679 \def\Fld@width{\DefaultWidthofChoiceMenu}
2680 \def\Fld@height{\DefaultHeightofChoiceMenu}
2681 \begingroup
2682 \Fld@menulength=0 %
2683 \@tempdima\z@
2684 \clist_map_variable:nN { #3 } \@curropt
2685 %\@for\@curropt:=#3\do
2686 {%
2687 \expandafter\Fld@checkequals\@curropt==\\%
2688 \Hy@StepCount\Fld@menulength
2689 \settowidth{\@tempdimb}{\@currDisplay}%
2690 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2691 }%
2692 \advance\@tempdima by~15\p@
2693 \begingroup
2694 \HyField@SetKeys{#1}
2695 \edef\x{\endgroup
2696 \noexpand\expandafter
2697 \noexpand\HyField@SetKeys
2698 \noexpand\expandafter{%
2699 \expandafter\noexpand\csname DefaultOptionsof%
2700 \ifFld@radio
2701 Radio%
2702 \else
2703 \ifFld@combo
2704 \ifFld@popdown
2705 PopdownBox%
2706 \else
2707 ComboBox%
2708 \fi
2709 \else
2710 ListBox%
2711 \fi
2712 \fi
2713 \endcsname
2714 }%

```

```

2715 } \x
2716 \HyField@SetKeys{#1}%
2717 \PDFForm@Name
2718 \ifFld@hidden\def\Fld@width{1sp}\fi
2719 \ifx\Fld@value\relax
2720 \let\Fld@value\Fld@default
2721 \fi
2722 \LayoutChoiceField{#2}{%
2723 \ifFld@radio
2724 \HyField@FlagsRadioButton
2725 \__hypRadio{#3}%
2726 \else
2727 \begingroup
2728 \HyField@FlagsChoice
2729 \ifdim\Fld@width<\@tempdima
2730 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2731 \edef\Fld@width{\the\@tempdima}%
2732 \fi
2733 \ifFld@combo
2734 \else
2735 \@tempdima=\the\Fld@menulength\Fld@charsize
2736 \advance\@tempdima by~\Fld@borderwidth bp %
2737 \advance\@tempdima by~\Fld@borderwidth bp %
2738 \edef\Fld@height{\the\@tempdima}%
2739 \fi
2740 \__hypListbox{#3}%
2741 \endgroup
2742 \fi
2743 }%
2744 \endgroup
2745 }
2746
2747 \def\__hypRadio#1{%
2748 \Fld@listcount=0~%
2749 %\show\Fld@default
2750 \EdefEscapeName\Fld@default{\Fld@default}%
2751 \clist_map_variable:nNn { #1 } \@curropt
2752 %\@for\@curropt:=#1\do
2753 {%
2754 \expandafter\Fld@checkequals\@curropt==\%
2755 \EdefEscapeName\@currValue{\@currValue}%
2756 \Hy@StepCount\Fld@listcount
2757 \@currDisplay\space
2758 \leavevmode
2759 \HyAnn@AbsPageLabel
2760 \Hy@escapeform\PDFForm@Radio
2761 \pdfxform_if_exist:nF { __hyp_xform_Ding }
2762 {
2763 \pdfxform_new:nnn { __hyp_xform_Ding } {}
2764 {
2765 \group_begin:
2766 \fontfamily{pzd}
2767 \fontencoding{U}
2768 \fontseries{m}

```

```

2769         \fontshape{n}
2770         \selectfont
2771         \char123
2772         \group_end:
2773     }
2774 }
2775 \pdfannot_box:nnne
2776 {\Fld@width}
2777 {\Fld@height}
2778 {0pt} %is this correct?
2779 {
2780     \PDFForm@Radio
2781     /AP
2782     <<
2783     /N
2784     <<
2785     /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2786     %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2787     >>
2788     >>
2789 }
2790 {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2791 \int_compare:nNnT { \Fld@listcount } = { 1 }
2792 { \HyField@AddToFields }
2793 \c_space_tl % deliberate space between radio buttons
2794 % to do: --> should be configurable
2795 }%
2796 }
2797
2798 \newcount\Fld@listcount
2799 \def\__hypListbox#1
2800 {
2801     \HyField@PDFChoices{#1}
2802     \mode_leave_vertical:
2803     \HyAnn@AbsPageLabel
2804     \Hy@escapeform\PDFForm@List
2805     \pdf_link_user:nnn
2806         {widget} %perhaps we need more types??
2807         {\PDFForm@List}
2808         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2809     \HyField@AddToFields
2810 }
2811
2812
2813 \def\@PushButton[#1]#2{% parameters, label
2814     \def\Fld@name{#2}%
2815     \group_begin:
2816         \exp_args:No\HyField@SetKeys
2817         {
2818             \DefaultOptionsofPushButton,#1
2819         }
2820         \PDFForm@Name
2821         \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2822         {

```

```

2823 \HyField@FlagsPushButton
2824 \legacy_if:nT {Fld@hidden}
2825 {
2826   \def\Fld@width{1sp}
2827 }
2828 \LayoutPushButtonField
2829 {
2830   \mode_leave_vertical:
2831   \HyAnn@AbsPageLabel
2832   \Hy@escapeform\PDFForm@Push
2833   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2834   \pdfannot_box:nnnn
2835     {\box_wd:N\l_tmpa_box}
2836     {\box_ht:N\l_tmpa_box}
2837     {\box_dp:N\l_tmpa_box} %is this correct?
2838     {\PDFForm@Push}
2839     {\box_use:N\l_tmpa_box}
2840   \HyField@AddToFields
2841 }
2842 }
2843 {
2844   \msg_error:nn { hyp }{ pdfa-no-push-button }
2845   \LayoutPushButtonField
2846   {
2847     \mode_leave_vertical:
2848     \MakeButtonField{#2}
2849   }
2850 }
2851 \group_end:
2852 }
2853
2854 \def\@Submit[#1]#2
2855 {
2856   \def\Fld@width {\DefaultWidthofSubmit}
2857   \def\Fld@height{\DefaultHeightofSubmit}
2858   \group_begin:
2859     \exp_args:No\HyField@SetKeys
2860     {
2861       \DefaultOptionsofSubmit,#1
2862     }
2863   \HyField@FlagsPushButton
2864   \HyField@FlagsSubmit
2865   \legacy_if:nT { Fld@hidden }
2866   {
2867     \def\Fld@width{1sp}
2868   }
2869   \mode_leave_vertical:
2870   \HyAnn@AbsPageLabel
2871   \Hy@escapeform\PDFForm@Submit
2872   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2873   \pdfxform_if_exist:nF
2874   { __hyp_xform_Submit }
2875   {
2876     \pdfxform_new:nnn { __hyp_xform_Submit }{ }

```

```

2877     {
2878         \fbox{\color_select:n{yellow}\textsf{Submit}}
2879     }
2880     \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2881     {
2882         \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2883     }
2884 }
2885 \pdfannot_box:nnnn
2886 {\box_wd:N\l_tmpa_box}
2887 {\box_ht:N\l_tmpa_box}
2888 {\box_dp:N\l_tmpa_box} %is this correct?
2889 {
2890     \PDFForm@Submit
2891     /AP<<
2892         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2893         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2894     >>
2895 }
2896 \HyField@AddToFields
2897 \box_use:N\l_tmpa_box
2898
2899 \group_end:
2900 }
2901
2902 \def\@Reset[#1]#2
2903 {
2904     \def\Fld@width {\DefaultWidthofReset}
2905     \def\Fld@height{\DefaultHeightofReset}
2906     \group_begin:
2907     \exp_args:No\HyField@SetKeys
2908     {
2909         \DefaultOptionsofReset,#1
2910     }
2911     \mode_leave_vertical:
2912     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2913     {
2914         \HyField@FlagsPushButton
2915         \legacy_if:nT { Fld@hidden }
2916         { \def\Fld@width{1sp} }
2917         \HyAnn@AbsPageLabel
2918         \Hy@escapeform\PDFForm@Reset
2919         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2920         \pdfannot_box:nnnn
2921         {\box_wd:N\l_tmpa_box}
2922         {\box_ht:N\l_tmpa_box}
2923         {\box_dp:N\l_tmpa_box} %is this correct?
2924         { \PDFForm@Reset }
2925         \HyField@AddToFields
2926         \box_use:N \l_tmpa_box
2927     }
2928     {
2929         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2930         \MakeButtonField{#2}

```

```

2931     }
2932   \group_end:
2933 }
2934
2935 \def\@CheckBox[#1]#2
2936 {% parameters, label
2937   \def\Fld@name{#2}
2938   \def\Fld@default{0}
2939   \group_begin:
2940     \def\Fld@width {\DefaultWidthofCheckBox}
2941     \def\Fld@height{\DefaultHeightofCheckBox}
2942     \exp_args:No\HyField@SetKeys
2943     {
2944       \DefaultOptionsofCheckBox,#1
2945     }
2946     \PDFForm@Name
2947     \HyField@FlagsCheckBox
2948     \legacy_if:nT { Fld@hidden }
2949     {
2950       \def\Fld@width{1sp}
2951     }
2952     \LayoutCheckField{#2}
2953     {
2954       \mode_leave_vertical:
2955       \HyAnn@AbsPageLabel
2956       \Hy@escapeform\PDFForm@Check
2957       \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2958       {
2959         \pdfxform_new:nnn
2960         {__hyp_xform_CheckMarkYes}{-}
2961         {
2962           \group_begin:
2963           \fontfamily{pzd}
2964           \fontencoding{U}
2965           \fontseries{m}
2966           \fontshape{n}
2967           \selectfont
2968           \char51
2969           \group_end:
2970         }
2971         \pdfxform_new:nnn
2972         {__hyp_xform_CheckMarkOff}{-}
2973         {
2974           \group_begin:
2975           \fontfamily{pzd}
2976           \fontencoding{U}
2977           \fontseries{m}
2978           \fontshape{n}
2979           \selectfont
2980           \phantom{\char51} %perhaps xetex needs some small glyph ..
2981           \group_end:
2982         }
2983       }
2984       \pdfannot_box:nnnn

```



```

2985         {\Fld@width}
2986         {\Fld@height}
2987         {Opt} %is this correct?
2988         {\PDFForm@Check}
2989     \HyField@AddToFields %check if this works with xelatex ...
2990 }
2991 \group_end:
2992 }
2993 \ExplSyntaxOff
2994
2995 %hm. Should a luatex driver use type1 fonts in fields????
2996 \ExplSyntaxOn
2997 \def\Hy@FormObjects
2998 {
2999     \pdf_object_new:n    {__hyp/Encoding/pdfdoc }
3000     \pdf_object_new:n    {__hyp/Font/ZaDb }
3001     \pdf_object_new:n    {__hyp/Font/Helv }
3002     \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3003     {
3004         /Type/Encoding
3005         /Differences[
3006             24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3007             /ring/tilde
3008             \c_space_tl
3009             39/quotesingle
3010             \c_space_tl
3011             96/grave %
3012             \iow_newline:
3013             128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3014             /fraction/guilsinglleft/guilsinglright/minus/perthousand
3015             /quotedblbase/quotedblleft/quotedblright/quoteleft
3016             /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3017             /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3018             \iow_newline:
3019             164/currency
3020             \c_space_tl
3021             166/brokenbar
3022             \c_space_tl
3023             168/dieresis/copyright/ordfeminine
3024             \c_space_tl
3025             172/logicalnot/.notdef/registered/macron/degree/plusminus
3026             /twosuperior/threesuperior/acute/mu
3027             \c_space_tl
3028             183/periodcentered/cedilla/onesuperior/ordmasculine
3029             \c_space_tl
3030             188/onequarter/onehalf/threequarters
3031             \iow_newline:
3032             192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3033             /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3034             /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Uacute
3035             /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3036             /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3037             /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3038             /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave

```

```

3039         /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3040         /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3041         /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3042     ]
3043 }
3044 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3045 {
3046     /Type/Font
3047     /Subtype/Type1
3048     /Name/ZaDb
3049     /BaseFont/ZapfDingbats
3050 }
3051 \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3052 {
3053     /Type/Font
3054     /Subtype/Type1
3055     /Name/Helv
3056     /BaseFont/Helvetica
3057     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3058 }
3059 \global\let\Hy@FormObjects\relax
3060 }
3061 \ExplSyntaxOff
3062 \providecommand*{\Fld@pageobjref}{}
3063 \ifcsname pdf@escapestring\endcsname
3064     \def\Hy@escapeform#1{%
3065         \ifHy@pdfescapeform
3066             \let\Hy@escapestring\pdfescapestring
3067         \else
3068             \let\Hy@escapestring\@firstofone
3069         \fi
3070     }%
3071     \Hy@escapeform{}%
3072 \else
3073     \let\Hy@escapestring\@firstofone
3074     \def\Hy@escapeform#1{%
3075         \ifHy@pdfescapeform
3076             \def\Hy@escapestring##1{%
3077                 \noexpand\Hy@escapestring{\noexpand##1}%
3078             }%
3079             \edef\Hy@temp{#1}%
3080             \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
3081             \def\Hy@escapestring##1{%
3082                 \@ifundefined{Hy@esc@\string##1}{%
3083                     ##1%
3084                     \ThisShouldNotHappen
3085                 }{%
3086                     \csname Hy@esc@\string##1\endcsname
3087                 }%
3088             }%
3089         \else
3090             \let\Hy@escapestring\@firstofone
3091         \fi
3092     }%

```

```

3093 \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3094   \ifx\#3\%
3095   \else
3096     \expandafter
3097     \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3098     \Hy@ReturnAfterFi{%
3099       \Hy__hypescapeform#3\@nil
3100     }%
3101   \fi
3102 }%
3103 \fi
3104 \def\PDFForm@Name{%
3105   \PDFForm__hypName\Fld@name
3106   \ifx\Fld@altname\relax
3107   \else
3108     \PDFForm__hypName\Fld@altname
3109   \fi
3110   \ifx\Fld@mappingname\relax
3111   \else
3112     \PDFForm__hypName\Fld@mappingname
3113   \fi
3114 }
3115 \def\PDFForm__hypName#1{%
3116   \begingroup
3117     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3118       \ifHy@unicode
3119         \Hy@unicodedefalse
3120       \fi
3121     \fi
3122     \pdfstringdef\Hy@gtemp#1%
3123   \endgroup
3124   \let#1\Hy@gtemp
3125 }
3126 \def\Fld@X@additionalactions{%
3127   \ifx\Fld@keystroke@code\@empty
3128   \else
3129     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3130   \fi
3131   \ifx\Fld@format@code\@empty
3132   \else
3133     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3134   \fi
3135   \ifx\Fld@validate@code\@empty
3136   \else
3137     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3138   \fi
3139   \ifx\Fld@calculate@code\@empty
3140   \else
3141     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3142   \fi
3143   \ifx\Fld@onfocus@code\@empty
3144   \else
3145     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3146   \fi

```

```

3147 \ifx\Fld@onblur@code\@empty
3148 \else
3149 /B1<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3150 \fi
3151 \ifx\Fld@onmousedown@code\@empty
3152 \else
3153 /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3154 \fi
3155 \ifx\Fld@onmouseup@code\@empty
3156 \else
3157 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3158 \fi
3159 \ifx\Fld@onenter@code\@empty
3160 \else
3161 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3162 \fi
3163 \ifx\Fld@onexit@code\@empty
3164 \else
3165 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3166 \fi
3167 }
3168 \ExplSyntaxOn
3169 \def\Fld@additionalactions
3170 {%
3171 \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{%
3172 {
3173 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3174 {/AA<<\Fld@X@additionalactions>>}
3175 }
3176 }
3177 \ExplSyntaxOff
3178 \def\Fld@annotnames{%
3179 /T(\Fld@name)%
3180 \ifx\Fld@altname\relax
3181 \else
3182 /TU(\Fld@altname)%
3183 \fi
3184 \ifx\Fld@mappingname\relax
3185 \else
3186 /TM(\Fld@mappingname)%
3187 \fi
3188 }
3189 \ExplSyntaxOn
3190 \def\PDFForm@Check
3191 {
3192 /Subtype/Widget
3193 ~\Fld@annotflags
3194 ~\Fld@pageobjref
3195 ~\Fld@annotnames
3196 /FT/Btn
3197 \Fld@flags
3198 /Q~\Fld@align
3199 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3200 /AP

```

```

3201     <<
3202     /N
3203     <<
3204         /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3205         /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3206     >>
3207 >>
3208 /MK<<
3209     \int_compare:nNnF {\Fld@rotation}={0}
3210     {
3211         /R~\Fld@rotation
3212     }
3213     \tl_if_empty:NF\Fld@bordercolor
3214     {
3215         /BC[\Fld@bordercolor]
3216     }
3217     \tl_if_empty:NF\Fld@bcolor
3218     {
3219         /BG[\Fld@bcolor]
3220     }
3221     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3222 >>
3223 /DA
3224 (
3225     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3226     \tl_if_empty:NF \Fld@color
3227     {
3228         \c_space_tl \Fld@color
3229     }
3230 )
3231 /H/P
3232 \legacy_if:nTF {\Fld@checked}
3233 {
3234     /V/Yes /AS/Yes
3235 }
3236 {
3237     /V/Off /AS/Off
3238 }
3239 \Fld@additionalactions
3240 }
3241 \ExplSyntaxOff
3242 \ExplSyntaxOn
3243 \def\PDFForm@Push
3244 {
3245     /Subtype/Widget
3246     ~\Fld@annotflags
3247     ~\Fld@pageobjref
3248     ~\Fld@annotnames
3249     /FT/Btn
3250     ~\Fld@flags
3251     /H/P
3252     /BS<</W~\Fld@borderwidth/S~\Fld@borderstyle>>
3253     \bool_if:nT
3254     {

```

```

3255         !\int_compare_p:nNn {\Fld@rotation} = {0}
3256         ||
3257         \tl_if_exist_p:N \Fld@bordercolor
3258     }
3259     {
3260         /MK
3261         <<
3262         \int_compare:nNf {\Fld@rotation} = {0}
3263         {
3264             /R~\Fld@rotation
3265         }
3266         \tl_if_exist:NT \Fld@bordercolor
3267         {
3268             /BC[\Fld@bordercolor]
3269         }
3270         >>
3271     }
3272     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3273     \Fld@additionalactions
3274 }
3275
3276 \ExplSyntaxOff
3277 \def\PDFForm@List{%
3278 /Subtype/Widget%
3279 \Fld@annotflags
3280 \Fld@pageobjref
3281 \Fld@annotnames
3282 /FT/Ch%
3283 \Fld@flags
3284 /Q \Fld@align
3285 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3286 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3287 \ifx\Fld@bordercolor\relax\else 1\fi
3288 \ifx\Fld@bcolor\relax \else 1\fi
3289 \space
3290 \else
3291 /MK<<%
3292 \ifnum\Fld@rotation=\z@
3293 \else
3294 /R \Fld@rotation
3295 \fi
3296 \ifx\Fld@bordercolor\relax
3297 \else
3298 /BC[\Fld@bordercolor]%
3299 \fi
3300 \ifx\Fld@bcolor\relax
3301 \else
3302 /BG[\Fld@bcolor]%
3303 \fi
3304 >>%
3305 \fi
3306 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3307 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3308 \Fld@choices

```

```

3309 \Fld@additionalactions
3310 }
3311 \ExplSyntaxOn
3312 \def\PDFForm@Radio
3313 {
3314   /Subtype/Widget
3315   ~\Fld@annotflags
3316   ~\Fld@pageobjref
3317   ~\Fld@annotnames
3318   /FT/Btn
3319   \Fld@flags
3320   /H/P
3321   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3322   /MK<<
3323     \ifnum\Fld@rotation=\z@
3324     \else
3325       /R~\Fld@rotation
3326     \fi
3327     \ifx\Fld@bordercolor\relax
3328     \else
3329       /BC[\Fld@bordercolor]%
3330     \fi
3331     \ifx\Fld@bcolor\relax
3332     \else
3333       /BG[\Fld@bcolor]%
3334     \fi
3335     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3336   >>
3337   /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3338     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3339   \ifx\Fld@default\@empty
3340     /V/Off%
3341     /DV/Off%
3342   \else
3343     /V/\Fld@default
3344     /DV/\Fld@default
3345   \fi
3346   \Fld@additionalactions
3347 }
3348 \ExplSyntaxOff
3349 \ExplSyntaxOn
3350 % Does an appeareance dict make sense here?
3351 \def\PDFForm@Text
3352 {
3353   /Subtype/Widget
3354   ~\Fld@annotflags
3355   ~\Fld@pageobjref
3356   ~\Fld@annotnames
3357   /FT/Tx
3358   ~\Fld@flags
3359   /Q~\Fld@align
3360   /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3361   \bool_if:nT
3362   {

```

```

3363         !\int_compare_p:nNn {\Fld@rotation} = {0}
3364         ||
3365         \tl_if_exist_p:N \Fld@bordercolor
3366         ||
3367         \tl_if_exist_p:N \Fld@bcolor
3368     }
3369     {
3370         /MK
3371         <<
3372         \int_compare:nNf {\Fld@rotation} = {0}
3373         {
3374             /R~\Fld@rotation
3375         }
3376         \tl_if_exist:NT \Fld@bordercolor
3377         {
3378             /BC[\Fld@bordercolor]
3379         }
3380         \tl_if_exist:NT \Fld@bcolor
3381         {
3382             /BG[\Fld@bcolor]
3383         }
3384         >>
3385     }
3386     /DA
3387     (
3388         /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3389         \tl_if_empty:NF {\c_space_tl\Fld@color}
3390     )
3391     /DV(\Hy@escapestring{\Fld@default})
3392     /V(\Hy@escapestring{\Fld@value})
3393     ~\Fld@additionalactions
3394     \int_compare:nNt { \Fld@maxlen}>{0}
3395     {
3396         /MaxLen~\Fld@maxlen
3397     }
3398 }
3399 \ExplSyntaxOff
3400
3401 \def\PDFForm@Submit{%
3402     /Subtype/Widget%
3403     \Fld@annotflags
3404     \Fld@pageobjref
3405     \Fld@annotnames
3406     /FT/Btn%
3407     \Fld@flags
3408     /H/P%
3409     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3410     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3411         \ifx\Fld@bordercolor\relax\else 1\fi
3412         \space
3413     \else
3414         /MK<<%
3415         \ifnum\Fld@rotation=\z@
3416         \else

```



```

3417         /R \Fld@rotation
3418     \fi
3419     \ifx\Fld@bordercolor\relax
3420     \else
3421         /BC[\Fld@bordercolor]%
3422     \fi
3423     >>%
3424 \fi
3425 /A<<%
3426     /S/SubmitForm%
3427     /F<<%
3428         /FS/URL%
3429         /F(\Hy@escapestring{\Form@action})%
3430     >>%
3431     \Fld@submitflags
3432 >>%
3433 \Fld@additionalactions
3434 }
3435 \ExplSyntaxOn
3436 \def\PDFForm@Reset{%
3437     /Subtype/Widget%
3438     \Fld@annotflags
3439     \Fld@pageobjref
3440     \Fld@annotnames
3441     /FT/Btn%
3442     \Fld@flags
3443     /H/P%
3444     /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3445     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3446         \ifx\Fld@bordercolor\relax\else 1\fi
3447         \space
3448     \else
3449         /MK<<%
3450         \ifnum\Fld@rotation=\z@
3451         \else
3452             /R~\Fld@rotation
3453         \fi
3454         \ifx\Fld@bordercolor\relax
3455         \else
3456             /BC[\Fld@bordercolor]%
3457         \fi
3458         >>%
3459     \fi
3460     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3461     /A<</S/ResetForm>>%
3462     \Fld@additionalactions
3463 }%
3464
3465
3466 %these patterns are used in hyperref checks.
3467 %it is unclear if they are really useful and if a backend support is
3468 %needed.
3469 \str_case:VnF \c_sys_backend_str
3470 {

```

```

3471 { pdfmode }
3472 {
3473   \def\HyPat@ObjRef
3474   {
3475     [0-9]*[1-9][0-9]*~0~R
3476   }
3477 }
3478 { dvipdfmx }
3479 {
3480   \def\HyPat@ObjRef
3481   {
3482     @[\~]+
3483   }
3484 }
3485 { xdvipdfmx }
3486 {
3487   \def\HyPat@ObjRef
3488   {
3489     @[\~]+
3490   }
3491 }
3492 }
3493 { %also set in hyperref sty, so probably not needed.
3494   \def\HyPat@ObjRef/{.+}
3495 }
3496
3497
3498 \ExplSyntaxOff
3499 % UF: removed Hy@writebookmark
3500 %   \Hy@currentbookmarklevel{0}
3501 %   \Hy@numberline
3502 %   \__hypwritetorep
3503 %   counter{bookmark@seq@number}
3504 % removed \HyPsd@SanitizeForOutFile, not needed
3505 % removed \currentpdfbookmark, defined by bookmark,
3506 % should use \newcommand there
3507 % removed \subpdfbookmark, defined by bookmark,
3508 % should use \newcommand there
3509 % removed \belowpdfbookmark, defined by bookmark,
3510 % should use \newcommand there
3511 % removed \pdfbookmark, defined by bookmark,
3512 % \BOOKMARK
3513 % \@BOOKMARK
3514 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3515 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3516 %% removed \ReadBookmarks / unneeded with bookmark.
3517 %% removed \Hy@OutlineName
3518 %% removed \check@bm@number
3519 %% removed \calc@bm@number
3520
3521 \ifHy@implicit
3522 \else
3523   \expandafter\endinput
3524 \fi

```

```

3525 \newlength\Hy@SectionHShift
3526 \def\Hy@SectionAnchorHref#1{%
3527   \ifx\protect\@typeset@protect
3528     \Hy__hypSectionAnchor{#1}%
3529   \fi
3530 }
3531 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3532   \leavevmode
3533   \hbox to 0pt{%
3534     \kern-\Hy@SectionHShift
3535     \Hy@raisedlink{%
3536       \hyper@anchorstart{#1}\hyper@anchorend
3537     }%
3538     \hss
3539   }%
3540 }
3541 \@ifundefined{hyper@nopatch@sectioning}
3542 {
3543   \let\H@old@ssect\@ssect
3544   \def\@ssect#1#2#3#4#5{%
3545     \Hy@MakeCurrentHrefAuto{section*}%
3546     \setlength{\Hy@SectionHShift}{#1}%
3547     \begingroup
3548       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3549       \toks\tw@\expandafter{%
3550         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3551         #5%
3552       }%
3553   \edef\x{\endgroup
3554     \the\toks@{\the\toks\tw@}%
3555   }\x
3556 }
3557 \let\H@old@schapter\@schapter
3558 \def\@schapter#1{%
3559   \begingroup
3560     \let\@mkboth\@gobbletwo
3561     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3562     \Hy@raisedlink{%
3563       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3564     }%
3565   \endgroup
3566   \H@old@schapter{#1}%
3567 }
3568 \@ifundefined{chapter}{-}{%
3569   \let\Hy@org@chapter\@chapter
3570   \def\@chapter{%
3571     \def\Hy@next{%
3572       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3573       \Hy@raisedlink{%
3574         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3575       }%
3576     }%
3577     \ifnum\c@secnumdepth>\m@ne
3578       \@ifundefined{if@mainmatter}%

```

```

3579 \iftrue{\csname if@mainmatter\endcsname}%
3580 \let\Hy@next\relax
3581 \fi
3582 \fi
3583 \Hy@next
3584 \Hy@org@chapter
3585 }%
3586 }
3587 \let\H@old@part\@part
3588 \begingroup\expandafter\expandafter\expandafter\endgroup
3589 \expandafter\ifx\csname chapter\endcsname\relax
3590 \let\Hy@secnum@part\z@
3591 \else
3592 \let\Hy@secnum@part\m@ne
3593 \fi
3594 \def\@part{%
3595 \ifnum\Hy@secnum@part>\c@secnumdepth
3596 \phantomsection
3597 \fi
3598 \H@old@part
3599 }
3600 \let\H@old@spart\@spart
3601 \def\@spart#1{%
3602 \Hy@MakeCurrentHrefAuto{part*}%
3603 \Hy@raisedlink{%
3604 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3605 }%
3606 \H@old@spart{#1}%
3607 }
3608 \let\H@old@sect\@sect
3609 \def\@sect#1#2#3#4#5#6[#7]#8{%
3610 \ifnum #2>\c@secnumdepth
3611 \expandafter\@firstoftwo
3612 \else
3613 \expandafter\@secondoftwo
3614 \fi
3615 {%
3616 \Hy@MakeCurrentHrefAuto{section*}%
3617 \setlength{\Hy@SectionHShift}{#3}%
3618 \begingroup
3619 \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3620 \toks\tw@\expandafter{%
3621 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3622 #8%
3623 }%
3624 \edef\x{\endgroup
3625 \the\toks@{\the\toks\tw@}%
3626 }\x
3627 }{%
3628 \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3629 }%
3630 }
3631 }{}
3632 \expandafter\def\csname Parent-4\endcsname{}

```

```

3633 \expandafter\def\csname Parent-3\endcsname{}
3634 \expandafter\def\csname Parent-2\endcsname{}
3635 \expandafter\def\csname Parent-1\endcsname{}
3636 \expandafter\def\csname Parent0\endcsname{}
3637 \expandafter\def\csname Parent1\endcsname{}
3638 \expandafter\def\csname Parent2\endcsname{}
3639 \expandafter\def\csname Parent3\endcsname{}
3640 \expandafter\def\csname Parent4\endcsname{}
3641 %%
3642 %% End of file 'hgeneric-testphase.def'.
3643 \</package>

3644 \*colorscheme>
3645 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3646 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3647 % linkcolor=
3648 %,filecolor=
3649 %,urlcolor=
3650 %,menucolor=
3651 %,runcolor=
3652 %,linkbordercolor=
3653 %,filebordercolor=
3654 %,urlbordercolor=
3655 %,menubordercolor=
3656 %,runbordercolor=
3657
3658 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3659 {
3660     linkcolor      = [rgb]{1,0,0}, %red
3661     filecolor      = [rgb]{0,1,1}, %cyan
3662     urlcolor       = [rgb]{1,0,1}, %magenta
3663     menucolor      = [rgb]{1, 0, 0}, %red
3664     runcolor       = [rgb]{0,1,1}, %cyan
3665     %-----
3666     linkbordercolor = [rgb]{1, 0 ,0 },
3667     filebordercolor = [rgb]{0, .5, .5},
3668     urlbordercolor  = [rgb]{0, 1, 1},
3669     menubordercolor = [rgb]{1, 0, 0},
3670     runbordercolor  = [rgb]{0, .7, .7}
3671 }
3672
3673 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3674 {
3675     linkcolor      = [rgb]{0,0.2,0.6},
3676     filecolor      = [rgb]{0.8,0,0.8},
3677     urlcolor       = [rgb]{0.8,0,0.8},
3678     menucolor      = [rgb]{0,0.2,0.6},
3679     runcolor       = [rgb]{0.8,0,0.8},
3680     %----- %-----
3681     linkbordercolor = [rgb]{0,0.2,0.6},
3682     filebordercolor = [rgb]{0.8,0,0.8},
3683     urlbordercolor  = [rgb]{0.8,0,0.8},
3684     menubordercolor = [rgb]{0,0.2,0.6},
3685     runbordercolor  = [rgb]{0.8,0,0.8}
3686 }

```

```

3687
3688 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3689 { %two colors: intern/extern
3690   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3691   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3692   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3693   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3694   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3695   %----- %-----
3696   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3697   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3698   urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3699   menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3700   runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3701 }
3702
3703 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3704 { %all darkgray
3705   linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3706   filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3707   urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3708   menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3709   runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3710   %----- %-----
3711   linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3712   filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3713   urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3714   menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3715   runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3716 }
3717
3718 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3719 { %dvipsnam.def
3720   linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3721   filecolor      = [rgb]{1, 0, 0}, %Red
3722   urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3723   menucolor      = [rgb]{1, 0, 0}, %Red
3724   runcolor       = [rgb]{1, 0, 0}, %Red
3725   %----- %-----
3726   linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3727   filebordercolor = [rgb]{1, 0, 0}, %Red
3728   urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3729   menubordercolor = [rgb]{1, 0, 0}, %Red
3730   runbordercolor  = [rgb]{1, 0, 0} %Red
3731 }
3732
3733 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3734 { %dvipsnam.def
3735   linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3736   filecolor      = [rgb]{0, 1, 0}, %Green
3737   urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3738   menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3739   runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3740   %----- %-----

```

```

3741     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3742     filebordercolor = [rgb]{0, 1, 0},    %Green
3743     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3744     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3745     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3746 }
3747
3748
3749 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3750 {
3751     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3752     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3753     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3754     menucolor      = [rgb]{0.44706, 0.45882, 0},
3755     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3756     %----- %-----
3757     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3758     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3759     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3760     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3761     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3762 }
3763
3764 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3765 {
3766     linkcolor      = [rgb]{0,0,0},
3767     filecolor      = [rgb]{0,0,0},
3768     urlcolor       = [rgb]{0,0,0},
3769     menucolor      = [rgb]{0,0,0},
3770     runcolor       = [rgb]{0,0,0},
3771     %----- %-----
3772     linkbordercolor = [rgb]{0,0,0},
3773     filebordercolor = [rgb]{0,0,0},
3774     urlbordercolor  = [rgb]{0,0,0},
3775     menubordercolor = [rgb]{0,0,0},
3776     runbordercolor  = [rgb]{0,0,0}
3777 }
3778 </colorscheme>

```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
\#	280, 758
\\$	279
\%	759
\-	2166
\.	524, 528, 530
@curropt commands:	
\@curropt:	2685, 2752
\[2166
\]	20, 21, 31, 32, 33, 41, 45, 55, 75, 82, 89, 96, 103, 118, 123, 124, 132, 133, 142, 143, 144, 145, 146, 153, 161, 281, 282, 908, 2687, 2754, 3094
_	522, 524, 528, 530
\]	2166
A	
\A	522, 2166
\Acrobatmenu	18, 174
\addcontentsline	13
\AddToDocumentProperties	389, 2161, 2262
\AddToHook	427, 440
\AddToHookNext	199
\advance	2617, 2692, 2736, 2737
allcolors (hypersetup key)	1027
\author	2
B	
\b	526
\begingroup	210, 278, 376, 2523, 2652, 2681, 2693, 2727, 3116, 3547, 3559, 3588, 3618
\belowpdfbookmark	3509
\bgroup	275, 376
\BOOKMARK	3512
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	268, 297, 356, 681, 704, 723, 743, 754, 789, 854, 919, 997, 1007, 1251, 1262
\bool_if:nTF	869, 1936, 1961, 3253, 3361
\bool_lazy_and:nnTF	424
\bool_lazy_or:nnTF	1281, 1330
\bool_new:N	204, 205, 508, 512, 516
\bool_set_true:N	517
bordercolormodel (hypersetup key)	13, 1048
box commands:	
\box_dp:N	618, 2837, 2888, 2923
\box_ht:N	617, 2836, 2887, 2922
\box_new:N	457, 519
\box_set_dp:Nn	2133
\box_set_ht:Nn	2140
\box_use:N	1268, 2839, 2897, 2926
\box_use_drop:N	1273
\box_wd:N	616, 2835, 2886, 2921
\l_tmpa_box	2833, 2835, 2836, 2837, 2839, 2872, 2886, 2887, 2888, 2897, 2919, 2921, 2922, 2923, 2926
C	
\catcode	279, 280
\char	2771, 2968, 2980
\chardef	159
\cite	33
clist commands:	
\clist_item:nn	2193
\clist_map_function:nN	125, 134
\clist_map_inline:nn	2326
\clist_map_variable:nNn	2684, 2751
color commands:	
\color_export:nnN	40, 432, 956, 1069
\color_select:n	964, 1000, 1272, 2878, 2882
\color_select:nn	970
\color_set:nn	2, 394, 981
\color_set:nnn	2, 393, 987
color names:	
hyp/annot/file	505
hyp/annot/link	505
hyp/annot/menu	505
hyp/annot/run	505
hyp/annot/url	505
colorfile (hypersetup key)	1027
colorlink (hypersetup key)	1027
colorlinks (hypersetup key)	1013
colormenu (hypersetup key)	1027
colorrun (hypersetup key)	1027
colorscheme (hypersetup key)	1, 1439
colorurl (hypersetup key)	1027
cs commands:	
\cs_generate_variant:Nn	156, 157, 158, 391, 456, 560, 786, 956, 973, 990
\cs_gset:Npn	1243
\cs_gset_eq:NN	410, 421
\cs_if_exist:NTF	13
\cs_if_exist_p:N	425

<code>\ExplSyntaxOn</code>	1264
.... 8, 11, 2555, 2612, 2636, 2996,	
3168, 3189, 3242, 3311, 3349, 3435	
extension (hypersetup key)	13, 1481
F	
<code>\fbox</code>	2790, 2878, 2882
<code>\fi</code>	7, 2511,
2538, 2541, 2550, 2650, 2658, 2659,	
2690, 2708, 2711, 2712, 2718, 2721,	
2730, 2732, 2739, 2742, 3069, 3091,	
3101, 3103, 3109, 3113, 3120, 3121,	
3130, 3134, 3138, 3142, 3146, 3150,	
3154, 3158, 3162, 3166, 3183, 3187,	
3286, 3287, 3288, 3295, 3299, 3303,	
3305, 3307, 3326, 3330, 3334, 3338,	
3345, 3410, 3411, 3418, 3422, 3424,	
3445, 3446, 3453, 3457, 3459, 3524,	
3529, 3581, 3582, 3593, 3597, 3614	
file (hypersetup key)	10, 1512
file commands:	
<code>\file_input:n</code>	9
fileborderstyle (hypersetup key)	14, 1141
filecolor (hypersetup key)	1027
final (hypersetup key)	1463
<code>\fontencoding</code>	2767, 2964, 2976
<code>\fontfamily</code>	2766, 2963, 2975
<code>\fontseries</code>	2768, 2965, 2977
<code>\fontshape</code>	2769, 2966, 2978
fp commands:	
<code>\fp_eval:n</code>	2091, 2119, 2135
G	
<code>\gdef</code>	281, 282
<code>\GetDocumentProperties</code>	3
<code>\global</code>	2617, 3059
group commands:	
<code>\group_begin:</code>	266, 295, 318,
335, 354, 550, 687, 696, 710, 716,	
748, 775, 794, 842, 859, 901, 924,	
950, 999, 1254, 1271, 2395, 2765,	
2815, 2858, 2906, 2939, 2962, 2974	
<code>\group_end:</code>	309, 326,
343, 371, 557, 692, 696, 728, 732,	
772, 775, 839, 842, 899, 901, 948,	
950, 1009, 1274, 1277, 2402, 2772,	
2851, 2899, 2932, 2969, 2981, 2991	
H	
<code>\hbox</code>	376, 3533
hbox commands:	
<code>\hbox_overlap_right:n</code>	1268
<code>\hbox_set:Nn</code>	2833, 2872, 2919
<code>\hbox_set:Nw</code>	1255
<code>\hbox_set_end:</code>	1264
<code>\hbox_set_to_wd:Nnn</code>	2117
hidefile (hypersetup key)	1415
hidelink (hypersetup key)	1415
hidelinks (hypersetup key)	1415
hidemenu (hypersetup key)	1415
hiderun (hypersetup key)	1415
hideurl (hypersetup key)	1415
hook commands:	
<code>\hook_gput_code:nnn</code>	
..... 403, 414, 993, 1003, 1247, 1258	
<code>\hook_new:n</code>	392, 535, 630
<code>\hook_new_pair:nn</code>	
..... 261, 289, 313, 330, 348	
<code>\hook_use:n</code>	
. 265, 285, 294, 310, 317, 327, 334,	
344, 353, 372, 552, 636, 645, 689, 712	
<code>\href</code>	4, 5, 19, 263
<code>\hreflaunch</code>	5
<code>\hrefpdf</code>	5, 19, 314, 384
<code>\hrefrun</code>	5, 331, 385
<code>\hrefurl</code>	5, 19, 291, 383
<code>\hss</code>	3538
Hy internal commands:	
<code>\Hy_hyescapeform</code> ..	3080, 3093, 3099
<code>\Hy_hypSectionAnchor</code> ...	3528, 3531
HyField internal commands:	
<code>\HyField_hypAddToFields</code> .	2517, 2544
hyp commands:	
<code>\l_hyp_annot_colorfile_bool</code> ...	506
<code>\l_hyp_annot_colorlink_bool</code> ...	506
<code>\l_hyp_annot_colormenu_bool</code> ...	506
<code>\l_hyp_annot_colorrrun_bool</code>	506
<code>\l_hyp_annot_colorurl_bool</code>	506
<code>\l_hyp_annot_ocgcolorfile_bool</code> .	510
<code>\l_hyp_annot_ocgcolorlink_bool</code> .	510
<code>\l_hyp_annot_ocgcolormenu_bool</code> .	510
<code>\l_hyp_annot_ocgcolorrrun_bool</code> ..	510
<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	510
<code>\l_hyp_current_dest_name_tl</code>	
..... 629, 688, 711	
hyp internal commands:	
<code>\g__hyp_AcroForm_CoFields_prop</code> ..	
..... 2481, 2499, 2568, 2570, 2585	
<code>\g__hyp_AcroForm_Fields_prop</code> ...	
..... 2482, 2492, 2563	
<code>\l__hyp_annot_GoTo_bool</code> 681, 704, 723	
<code>\l__hyp_annot_GoTo_bool\l__-</code>	
_hyp_annot_URI_bool\l__-	
hyp_annot_GoToR_bool\l__-	
hyp_annot_Named_bool\l__-	
hyp_annot_Launch_bool	514
<code>\l__hyp_annot_GoToR_bool</code>	789
<code>\l__hyp_annot_Launch_bool</code>	854

\l__hyp_annot_Named_bool	919	\l__hyp_href_pdf_destination_tl	207, 218, 257, 325
\c__hyp_annot_types_seq	469, 506, 510, 1027, 1328, 1426	\l__hyp_href_pdf_page_tl 208, 226, 819	
\l__hyp_annot_URI_bool	743	__hyp_href_run_aux:nn	337, 340
\g__hyp_bordercolormodel_str	434, 505, 1052, 1071	\l__hyp_href_run_parameter_tl	209, 230, 342
__hyp_check_link_nesting:TF	590, 598, 600, 684, 707, 725, 746, 792, 857, 922	__hyp_href_url_aux:n	363, 366
__hyp_citebordercolor_hook_init:	399, 412, 421	__hyp_href_url_aux:nn	304, 306
__hyp_citecolor_hook_init:	397, 401, 410	\l__hyp_href_url_encode_bool	204, 215, 253, 268, 297, 356
__hyp_clist_display:n	118, 125, 134	__hyp_href_url_format:	210, 216, 258, 369
__hyp_color_select:n 40, 957, 957, 973		\l__hyp_href_url_ismap_bool	205, 229, 754
__hyp_color_select_aux:wn	957, 961, 968	\l__hyp_href_url_protocol_tl	206, 217, 256, 308, 370
__hyp_color_set:nn	397, 399, 431, 974, 974, 990, 1032, 1068	__hyp_if_outer_link:	584
__hyp_color_set_aux:nwn 974, 978, 985		__hyp_if_outer_link:TF	600
\c__hyp_colorscheme_daleif_prop 3673		__hyp_link_goto_begin:nw	652, 690, 713
\c__hyp_colorscheme_henryford_prop	3764	__hyp_link_goto_end:	674, 691, 727
\c__hyp_colorscheme_julian_prop 3688		\g__hyp_linknestlevel_int	583, 586, 683, 698, 706, 734, 745, 777, 791, 844, 856, 902, 921, 951
\c__hyp_colorscheme_phelype_prop	3749	g__hyp_linknestlevel_int	583
\c__hyp_colorscheme_szabolcsA_prop	3718	\c__hyp_map_annot_hyp_prop	469
\c__hyp_colorscheme_szabolcsB_prop	3733	\c__hyp_map_hyp_annot_prop	469, 991, 1055, 1095, 1122, 1130, 1141, 1168, 1176, 1245, 1353, 1389, 1399, 1512
\c__hyp_colorscheme_tivv_prop	3703	__hyp_ocg_init: 1187, 1187, 1243, 1253	
\l__hyp_dest_box	32, 519, 616, 617, 618, 2117, 2133, 2140	\l__hyp_optlang_regex 2165, 2166, 2195	
\l__hyp_dest_name_tmpa_tl	462, 655, 656, 662, 666, 668, 671, 812, 825	l__hyp_page/Trans	533
\l__hyp_dest_pdfremotestartview_tl	488, 820, 1924, 1928	__hyp_PageLabels_gpush: 567, 567, 580	
\g__hyp_dest_pdfstartpage_tl	488, 1935, 1937, 1944, 1962, 1969	\l__hyp_para_tmpa_tl 465, 868, 871, 882	
\g__hyp_dest_pdfstartview_tl	488, 1937, 1944, 1955, 1959, 1962, 1969	\l__hyp_para_tmpa_tl\l__hyp_text_tmpa_str\l__hyp_text_tmpa_str	462
\l__hyp_dest_pdfview_tl	504, 634, 643, 2086, 2089, 2096, 2099, 2100, 2101, 2102, 2103, 2104, 2109, 2113, 2149	__hyp_property_record:nn	449, 450, 456, 2621
\c__hyp_dest_startview_regex	520, 1922, 1953	__hyp_seconddoftwowithopt:wnn	382, 383, 384, 385
\c__hyp_dest_undefined_tl	468, 661, 662	__hyp_setup_info_date_key:nn	2248, 2281, 2282
__hyp_destination:nn	32, 603, 603, 634, 643	__hyp_setup_info_key:nn	2167, 2241, 2242, 2243, 2244, 2246, 2247
\l__hyp_filename_tmpa_tl	462, 795, 797, 802, 803, 808, 863, 864, 882	__hyp_store_metadata:nn	386, 391, 1531, 2209, 2227, 2261, 2276, 2285, 2301, 2369
__hyp_href_pdf_aux:nn	320, 323	__hyp_text_cleanup:N	540, 540, 554
		\l__hyp_text_enc_dest_print_tl	491, 811
		\l__hyp_text_enc_dest_tl	32, 491, 610, 667

\l__hyp_text_enc_file_print_tl ..	colorscheme	1 , 1439
..... 491 , 862	colorurl	1027
\l__hyp_text_enc_info_print_tl ..	debug	1463
..... 491 , 565	destlabel	13
\l__hyp_text_enc_para_print_tl ..	draft	1463
..... 491 , 867	extension	13 , 1481
\l__hyp_text_enc_uri_print_tl ...	file	10 , 1512
..... 270 ,	fileborderstyle	14 , 1141
273 , 299 , 302 , 358 , 361 , 491 , 751 , 1523	filecolor	1027
__hyp_text_pdfstring:nnN	final	1463
..... 548 , 548 , 560 , 562 ,	hidefile	1415
565 , 608 , 665 , 749 , 809 , 860 , 865 , 1523	hidelink	1415
__hyp_text_pdfstring_info:nn ..	hidelinks	1415
.... 563 , 563 , 2198 , 2202 , 2221 , 2231	hidemenu	1415
__hyp_text_purify:nn . 536 , 536 , 553	hiderun	1415
__hyp_text_string_from_unicode:nn	hideurl	1415
..... 544 , 544 , 555	hypertextnames	1481
\g__hyp_text_tmpa_str . 467 , 556 , 558	link	10 , 1512
\l__hyp_text_tmpa_str	linkborder	14
..... 466 , 553 , 554 , 555 , 556	linkborderstyle	14 , 1141
\l__hyp_tmpa_box 457 , 1255 , 1268 , 1273	linkcolor	1027
\l__hyp_tmpa_int	linkfileprefix	1481
457	linktoc	1481
\l__hyp_tmpa_seq	linktocpage	1481
.... 457 , 1922 , 1924 , 1953 , 1955 ,	localanchorname	1481
2076 , 2077 , 2081 , 2083 , 2107 , 2114 ,	menu	10 , 1512
2115 , 2116 , 2125 , 2127 , 2137 , 2142 ,	menuborder	14
2195 , 2196 , 2202 , 2572 , 2574 , 2580	menuborderstyle	14 , 1141
\l__hyp_tmpa_str	menucolor	1027
.... 457 , 2198 , 2202 , 2204 , 2206 ,	naturalnames	1481
2221 , 2222 , 2224 , 2231 , 2232 , 2236	nested-links	10
\l__hyp_tmpa_tl	nesting	14
.. 457 , 611 , 615 , 623 , 1072 , 1076 ,	ocgcolorfile	1281
1523 , 1524 , 1529 , 1920 , 1922 , 1951 ,	ocgcolorlink	1281
1953 , 2083 , 2084 , 2091 , 2193 , 2195	ocgcolorlinks	1281
\l__hyp_uri_tmpa_tl ... 462 , 752 , 753	ocgcolormenu	1281
hyp/anchor	ocgcolorrun	1281
hyp/annot/file (color name)	ocgcolorurl	1281
505	pageanchor	1481
hyp/annot/link (color name)	pdfauthor	2165
505	pdfborder	14
hyp/annot/menu (color name)	pdfborderstyle	14 , 1141
505	pdfcreationdate	14 , 2248
hyp/annot/run (color name)	pdfcreator	2165
505	pdfencoding	1453
hyp/annot/url (color name)	pdfinfo	2315
505	pdfkeywords	2165
hyp/text/pdfstring	pdflang	14 , 2154
535	pdflinkmargin	14
\hypercalcbp	pdfmetadate	14 , 2248
12 , 18 , 162	pdfmoddate	14 , 2248
\HyperDestNameFilter	pdfproducer	2165
13 , 609 , 666	pdfremotestartview	12
\hypersetup 1 , 2 , 5 , 9 , 10 , 13 , 19 , 68 , 100 , 181	pdfstartview	12
\hypersetup keys:		
allcolors		1027
bookmarkstype		13
bordercolormodel		13 , 1048
colorfile		1027
colorlink		1027
colorlinks		1013
colormenu		1027
colorrun		1027

menuborderstyle (hypersetup key)	14 , 1141
menucolor (hypersetup key)	1027
mode commands:	
\mode_if_horizontal:TF	605, 627
\mode_leave_vertical:	264, 293, 316, 333, 352, 654, 761, 827, 884, 927, 2802, 2830, 2847, 2869, 2911, 2954
msg commands:	
\msg_error:nn	426, 2844, 2929
\msg_info:nnn	2179, 2184
\msg_line_context:	83
\g_msg_module_name_prop	12
\msg_new:nnn	51, 58, 63, 67, 71, 78, 85, 92, 99, 106, 112, 119, 128, 138, 149
\msg_new:nnnn	15, 26, 37
\msg_warning:nn	171, 1459
\msg_warning:nnn	180, 658, 945
\msg_warning:nnnn	1303, 1344, 1557, 1601, 1635, 1658, 1720, 1749, 1783, 1798, 1811, 1824, 1849, 1884, 1900, 1927, 1958, 1991, 2005, 2032, 2066, 2148
\msg_warning:nnnnn	191, 1375, 1409, 1501, 1577, 1614, 1701, 1734, 1759, 1835, 1862, 1913, 2018, 2045, 2309, 2413, 2432, 2451, 2462
N	
naturalnames (hypersetup key)	1481
nested-links (hypersetup key)	10
nesting (hypersetup key)	14
\newcommand	165, 3506, 3508, 3510
\newcount	2613, 2798
\NewDocumentCommand	2475
\NewExpandableDocumentCommand	382
\newlength	3525
\noexpand	2696, 2697, 2698, 2699, 3077
\nolinkurl	4
O	
ocgcolorfile (hypersetup key)	1281
ocgcolorlink (hypersetup key)	1281
ocgcolorlinks (hypersetup key)	1281
ocgcolormenu (hypersetup key)	1281
ocgcolorrun (hypersetup key)	1281
ocgcolorurl (hypersetup key)	1281
P	
pageanchor (hypersetup key)	1481
\paperwidth	3
\PassOptionsToPackage	447, 1473, 1478
pdf commands:	
\pdf_bdcobject:nn	1267, 1270
\pdf_destination:nn	26, 156, 622
\pdf_destination:nnnn	615
\pdf_emc:	1269, 1275
\pdf_link_user:nnn	2805
\pdf_name_from_unicode_e:n	246, 799, 929
\pdf_object_if_exist:nTF	797
\pdf_object_new:n	1189, 1190, 1191, 1192, 2999, 3000, 3001
\pdf_object_ref:n	157, 808, 1195, 1197, 1219, 1220, 1223, 1227, 1232, 1237, 1242, 2590, 2592, 3057
\pdf_object_ref_last:	829, 2401
\pdf_object_unnamed_write:nn	828, 2397
\pdf_object_write:nnn	1193, 1199, 1209, 1221, 3002, 3044, 3051
\pdf_pageobject_ref:n	158, 1944, 1969, 2629
\pdf_string_from_unicode:nnN	546
\pdf_version:	1605, 1724, 1753, 1787, 1802, 1828, 1853, 1888, 1904, 2009, 2036
\pdf_version_compare:NnTF	1595, 1686, 1708, 1744, 1777, 1792, 1818, 1843, 1871, 1894, 1999, 2026
\pdf_version_compare_p:Nn	873, 1282, 1331
\pdf_version_major:	168, 1283, 1306, 1332, 1348
\pdf_version_minor:	167, 1306, 1348
pdfannot commands:	
\pdfannot_box:nnnn	2664, 2775, 2834, 2885, 2920, 2984
\pdfannot_box_ref_last:	2519, 2535
\pdfannot_dict_put:nnn	762, 829, 885, 930, 1073, 1108, 1132, 1154, 1178, 1360, 1391
\pdfannot_dict_remove:nn	1063, 1103, 1124, 1149, 1170, 1368, 1401
\pdfannot_link:nnn	763, 830, 886, 931
\pdfannot_link_goto_begin:nw	671
\pdfannot_link_goto_end:	676
\pdfannot_link_margin:n	8, 1639
\c_pdfannot_link_types_seq	514
pdfauthor (hypersetup key)	2165
\pdfbookmark	3511
pdfborder (hypersetup key)	14
pdfborderstyle (hypersetup key)	14 , 1141
pdfcreationdate (hypersetup key)	14 , 2248
pdfcreator (hypersetup key)	2165
\pdfdest	11
pdfdict commands:	
\pdfdict_new:n	533, 737, 782, 848, 913

\pdfdict_put:nnn	238, 245, 534, 738, 739, 753, 756, 783, 784, 799, 805, 815, 825, 849, 850, 864, 879, 914, 915, 928, 1665, 1666, 1670, 1671, 2410, 2420, 2424, 2427, 2429, 2448, 2457, 2459
\pdfdict_remove:nn	235, 876, 1675, 1676
\pdfdict_use:n	762, 828, 885, 891, 930, 935, 2399
pdfencoding (hypersetup key)	1453
\pdfscapestring	3066
pdffile commands:	
\pdffile_embed_file:nnn	786, 800
PDFForm internal commands:	
\PDFForm_hypName	3105, 3108, 3112, 3115
pdfinfo (hypersetup key)	2315
pdfkeywords (hypersetup key)	2165
pdflang (hypersetup key)	14, 2154
pdflinkmargin (hypersetup key)	14
pdfmanagement commands:	
\pdfmanagement_add:nn	1648
\pdfmanagement_add:nnn	569, 1219, 1220, 1242, 1529, 1549, 1565, 1569, 1589, 1597, 1626, 1681, 1691, 1715, 1729, 1741, 1746, 1771, 1779, 1794, 1820, 1845, 1879, 1896, 1942, 1967, 1981, 2001, 2028, 2057, 2160, 2180, 2185, 2206, 2224, 2235, 2259, 2274, 2296, 2382, 2401, 2565, 2582, 2589, 2591, 2593, 2599
\pdfmanagement_if_active_p:	425
\pdfmanagement_remove:nn	1526, 1545, 1553, 1573, 1585, 1610, 1622, 1630, 1644, 1653, 1697, 1712, 1731, 1768, 1807, 1832, 1858, 1875, 1909, 1939, 1964, 1977, 1986, 2014, 2041, 2053, 2062, 2189, 2218, 2256, 2271, 2379, 2392, 2603
\pdfmanagement_show:n	2566
pdfmeta commands:	
\pdfmeta_standard_verify:nnTF	925, 2821, 2912
\pdfmeta_standard_verify:nTF	2595, 3173
pdfmetadate (hypersetup key)	14, 2248
pdfmoddate (hypersetup key)	14, 2248
pdfproducer (hypersetup key)	2165
pdfremotestartview (hypersetup key)	12
pdfstartview (hypersetup key)	12
\pdfstringdef	8, 9, 23, 28, 3122
\pdfstringdefDisableCommands	383, 384, 385
pdfsubject (hypersetup key)	2165
pdftitle (hypersetup key)	2165
pdftrapped (hypersetup key)	2287
pdfversion (hypersetup key)	1453
pdfview (hypersetup key)	12, 2072
pdfxform commands:	
\pdfxform_if_exist:nTF	2761, 2873, 2957
\pdfxform_new:nnn	2477, 2763, 2876, 2880, 2959, 2971
\pdfxform_ref:n	2785, 2786, 2892, 2893, 3204, 3205
\phantom	2980
\phantomsection	13, 19, 3596
plainpages (hypersetup key)	1481
prg commands:	
\prg_do_nothing:	410, 421
\prg_generate_conditional_variant:Nnn	2624
\prg_new_conditional:Npnn	584
\prg_return_false:	587
\prg_return_true:	588
prop commands:	
\prop_const_from_keyval:Nn	471, 479, 3658, 3673, 3688, 3703, 3718, 3733, 3749, 3764
\prop_gput:Nnn	12, 2492, 2499
\prop_if_empty:NTF	2568
\prop_item:Nn	2585
\prop_map_inline:Nn	991, 1055, 1095, 1122, 1130, 1141, 1168, 1176, 1245, 1353, 1389, 1399, 1443, 1512, 2563, 2570
\prop_new:N	2481, 2482
property commands:	
\property_if_recorded:nn	2624
\property_if_recorded:nnTF	2627
\property_record:nn	453
\property_ref:nn	2631
\property_ref_undefined_warn:nn	2622
\protect	700, 767, 779, 835, 846, 895, 904, 953, 3527
\providecommand	163, 164, 2506, 2509, 2674, 3062
\ProvidesFile	3
R	
\ReadBookmarks	3516
\refstepcounter	12
regex commands:	
\regex_const:Nn	520
\regex_extract_once:NnN	2195
\regex_extract_once:NnNTF	1922, 1953
\regex_new:N	2165
\regex_set:Nn	2166

<code>\relax</code>	700, 767, 779, 835, 846, 895, 904, 953, 2677, 2678, 2719, 3059, 3106, 3110, 3180, 3184, 3287, 3288, 3296, 3300, 3327, 3331, 3411, 3419, 3446, 3454, 3580, 3589	<code>\str_lowercase:n</code>	
<code>\RemoveFromHook</code>	1535	2077, 2227, 2276, 2299, 2304
<code>\RenewDocumentCommand</code>	174	<code>\str_new:N</code>	461, 466, 467, 505
<code>\RequirePackage</code>	6, 32, 442, 3514	<code>\str_set:Nn</code>	538
<code>run</code> (hypersetup key)	10, 1512	<code>\str_set_convert:Nnnn</code>	8
<code>runborder</code> (hypersetup key)	14	<code>\str_set_eq:NN</code>	558
<code>runborderstyle</code> (hypersetup key)	14, 1141	<code>\str_tail:n</code>	2299, 2304
<code>runcolor</code> (hypersetup key)	1027	<code>\str_uppercase:n</code>	2292, 2298, 2303
S		<code>\string</code>	2506, 2509, 2530, 2548, 3082, 3086, 3097
		<code>\subpdfbookmark</code>	3507
<code>\selectfont</code>	2770, 2967, 2979	sys commands:	
seq commands:		<code>\c_sys_backend_str</code>	3469
<code>\seq_const_from_clist:Nn</code>	469	T	
<code>\seq_count:N</code>	2081, 2107	TeX and L ^A T _E X 2 _ε commands:	
<code>\seq_get_right:NN</code>	2083	<code>\@BOOKMARK</code>	3513
<code>\seq_if_empty:NTF</code>	2196	<code>\@CheckBox</code>	2935
<code>\seq_item:Nn</code>	1924, 1955, 2077, 2125, 2127, 2137, 2142, 2202	<code>\@ChoiceMenu</code>	2675
<code>\seq_map_inline:Nn</code>	506, 510, 514, 1027, 1328, 1426, 2580	<code>\@Form</code>	2557
<code>\seq_new:N</code>	459	<code>\@PushButton</code>	2813
<code>\seq_put_right:Nn</code>	2114, 2115, 2116, 2572	<code>\@Reset</code>	2902
<code>\seq_set_split:Nnn</code>	2076	<code>\@Submit</code>	2854
<code>\seq_sort:Nn</code>	2574	<code>\@TextField</code>	2640
<code>\setbox</code>	376	<code>\@bookmarksopenlevel</code>	1540
<code>\setlength</code>	3546, 3617	<code>\@bsphack</code>	452
<code>\setpdflinkmargin</code>	8	<code>\@chapter</code>	3569, 3570
<code>\settowidth</code>	2689	<code>\@currDisplay</code>	2689, 2757
<code>\show</code>	2749	<code>\@currValue</code>	2755, 2785
sort commands:		<code>\@currentHref</code>	
<code>\sort_return_same:</code>	2578	3550, 3563, 3574, 3604, 3621
<code>\sort_return_swapped:</code>	2577	<code>\@curropt</code>	2674, 2684, 2687, 2751, 2754
<code>\space</code>	2757, 3289, 3306, 3307, 3337, 3338, 3412, 3444, 3447	<code>\@empty</code>	2484, 2485, 2521, 2608, 2609, 2610, 2642, 2643, 2659, 3127, 3131, 3135, 3139, 3143, 3147, 3151, 3155, 3159, 3163, 3307, 3338, 3339
<code>\spacefactor</code>	605, 627	<code>\@endForm</code>	2608
<code>\stockwidth</code>	3	<code>\@esphack</code>	454
str commands:		<code>\@firstofone</code>	2512, 3068, 3073, 3090
<code>\c_colon_str</code>	908	<code>\@firstoftwo</code>	3611
<code>\c_hash_str</code>	758	<code>\@for</code>	2685, 2752
<code>\c_percent_str</code>	759	<code>\@gobbletwo</code>	3560
<code>\str_case:nnTF</code>	2175, 3469	<code>\@hyper@launch</code>	39, 908
<code>\str_case_e:nnTF</code>	2077	<code>\@ifnextchar</code>	275
<code>\str_compare:nNnTF</code>	2576	<code>\@ifundefined</code>	3082, 3541, 3568, 3578
<code>\str_gset:Nn</code>	1052	<code>\@mainaux</code>	2505, 2508, 2528, 2547
<code>\str_gset_eq:NN</code>	556	<code>\@mkboth</code>	3560
<code>\str_head:n</code>	2298, 2303	<code>\@ne</code>	2617
<code>\str_if_eq:nnTF</code>	612, 2204, 2222, 2232, 3171	<code>\@nil</code>	3080, 3093, 3099
<code>\str_if_eq_p:nn</code>	871, 1283, 1332	<code>\@part</code>	3587, 3594
		<code>\@pdfauthor</code>	23
		<code>\@pdfborder</code>	163
		<code>\@pdfborderstyle</code>	164
		<code>\@savsf</code>	605, 627

\@schapter	3557, 3558	\Fld@menulength	2682, 2688, 2735
\@secondoftwo	3613	\Fld@name 2641, 2676, 2814, 2937, 3105, 3179
\@sect	3608, 3609	\Fld@onblur@code	3147, 3149
\@spart	3600, 3601	\Fld@onclick@code	3272
\@ssect	3543, 3544	\Fld@onenter@code	3159, 3161
\@tempdima .	2683, 2690, 2692, 2729, 2730, 2731, 2735, 2736, 2737, 2738	\Fld@onexit@code	3163, 3165
\@tempdimb	2689, 2690	\Fld@onfocus@code	3143, 3145
\@typeset@protect	3527	\Fld@onmousedown@code ...	3151, 3153
\BKM@color	435	\Fld@onmouseup@code	3155, 3157
\c@secnumdepth	3577, 3595, 3610	\Fld@pageobjref 2610, 2625, 3062, 3194, 3247, 3280, 3316, 3355, 3404, 3439
\calc@bm@number	3519	\Fld@radiosymbol	3335
\check@bm@number	3518	\Fld@rotation	3209, 3211, 3255, 3262, 3264, 3286, 3292, 3294, 3323, 3325, 3363, 3372, 3374, 3410, 3415, 3417, 3445, 3450, 3452
\define@key	429	\Fld@submitflags	3431
\Fld@additionalactions	3169, 3239, 3273, 3309, 3346, 3393, 3433, 3462	\Fld@validate@code	3135, 3137
\Fld@align	3198, 3284, 3359	\Fld@value 2643, 2659, 2678, 2719, 2720, 3392
\Fld@altname ..	3106, 3108, 3180, 3182	\Fld@width	2644, 2658, 2665, 2669, 2679, 2718, 2729, 2731, 2776, 2790, 2808, 2826, 2856, 2867, 2904, 2916, 2940, 2950, 2985
\Fld@annotflags	3193, 3246, 3279, 3315, 3354, 3403, 3438	\Fld@X@additionalactions 3126, 3171, 3174
\Fld@annotnames	3178, 3195, 3248, 3281, 3317, 3356, 3405, 3440	\Form@action	3429
\Fld@bcolor	3217, 3219, 3300, 3302, 3331, 3333, 3367, 3380, 3382	\H@old@part	3587, 3598
\fld@bcolor	3288	\H@old@schapter	3557, 3566
\Fld@bordercolor	3213, 3215, 3257, 3266, 3268, 3287, 3296, 3298, 3327, 3329, 3365, 3376, 3378, 3411, 3419, 3421, 3446, 3454, 3456	\H@old@sect	3608, 3619, 3628
\Fld@borderstyle	3199, 3252, 3285, 3321, 3360, 3409, 3460	\H@old@spart	3600, 3606
\Fld@borderwidth .	2736, 2737, 3199, 3252, 3285, 3321, 3360, 3409, 3460	\H@old@ssect	3543, 3548
\Fld@calculate@code .	2521, 3139, 3141	\href@	275, 281
\Fld@calculate@sortkey	2532	\href@split	281, 282
\Fld@cbsymbol	3221	\Hy@abspage	579
\Fld@charsize 2735, 3225, 3306, 3337, 3388, 3444	\Hy@activeanchorfalse	650
\Fld@checkequals	2687, 2754	\Hy@activeanchortrue	641
\Fld@choices	3308	\Hy@AtBeginDocument	2486, 2502
\Fld@color	3226, 3228, 3307, 3338, 3389	\Hy@bookmarkstype	1541
\Fld@default 2642, 2659, 2677, 2720, 2749, 2750, 2938, 3339, 3343, 3344, 3391	\Hy@chapapp	3561, 3572
\Fld@flags	3197, 3250, 3283, 3319, 3358, 3407, 3442	\Hy@colorlink	34
\Fld@format@code	3131, 3133	\Hy@currentbookmarklevel	3500
\Fld@height	2645, 2666, 2669, 2680, 2738, 2777, 2790, 2808, 2857, 2905, 2941, 2986	\Hy@DisableOption	173
\Fld@keystroke@code	3127, 3129	\Hy@drafttrue	1472
\Fld@listcount	2748, 2756, 2791, 2798	\Hy@escapeform 2663, 2760, 2804, 2832, 2871, 2918, 2956, 3064, 3071, 3074
\Fld@mappingname	3110, 3112, 3184, 3186	\Hy@escapestring	3066, 3068, 3073, 3076, 3077, 3080, 3081, 3090, 3093, 3129, 3133, 3137, 3141, 3145, 3149, 3153, 3157, 3161, 3165, 3221, 3272, 3335, 3391, 3392, 3429
\Fld@maxlen	3394, 3396		

\Hy@finaltrue	1477	\HyField@cofields	2485
\Hy@FormObjects	2562, 2997, 3059	\HyField@FlagsCheckBox	2947
\Hy@gttemp	3122, 3124	\HyField@FlagsChoice	2728
\Hy@href	275	\HyField@FlagsPushButton	
\Hy@href@nextactionraw	239		2823, 2863, 2914
\Hy@href@page	227	\HyField@FlagsRadioButton	2724
\Hy@linkfileprefix	1486	\HyField@FlagsSubmit	2864
\Hy@linktoc	1497	\HyField@FlagsText	2657
\Hy@MakeCurrentHref	200	\HyField@PDFChoices	2801
\Hy@MakeCurrentHrefAuto		\HyField@SetKeys	2653, 2694,
	3545, 3561, 3572, 3602, 3616		2697, 2716, 2816, 2859, 2907, 2942
\Hy@next	3571, 3580, 3583	\HyPat@ObjRef	3473, 3480, 3487, 3494
\Hy@numberline	166, 3501	\hyper@link	283
\Hy@org@chapter	3569, 3584	\hyper@anchor	629
\Hy@OutlineName	3517	\hyper@anchorend	
\Hy@OutlineRerunCheck	3515		629, 3536, 3563, 3574, 3604
\Hy@pdfmajorversion	168	\hyper@anchorstart	
\Hy@pdfminorversion	167		629, 3536, 3563, 3574, 3604
\Hy@pdfstringtrue	29, 551	\hyper@link	34, 679
\Hy@pdfversion	3117	\hyper@linkend	34, 721
\Hy@pstringdef	561, 3097	\hyper@linkfile	325, 787
\Hy@PutCatalog	567	\hyper@linklaunch	37, 342, 852, 911
\Hy@raisedlink	3535, 3562, 3573, 3603	\hyper@linknamed	39, 176, 917
\Hy@RestoreLastskip	626	\hyper@linkstart	34, 702
\Hy@ReturnAfterFi	7, 3098	\hyper@linkurl	308, 369, 741
\Hy@safe@activestru	760, 2524	\hyper@normalise	
\Hy@SaveLastskip	606		275, 304, 320, 337, 363, 376
\Hy@secnum@part	3590, 3592, 3595	\HyPL@Labels	569, 579
\Hy@SectionAnchorHref	3526, 3550, 3621	\HyPL@storePageLabel	567
\Hy@SectionHShift		\HyPsd@SanitizeForOutFile	3504
	3525, 3534, 3546, 3617	\if@filesw	2504, 2527, 2546
\Hy@StepCount	2688, 2756	\ifFld@combo	2703, 2733
\Hy@temp	2525, 2540, 3079, 3080	\ifFld@hidden	2658, 2718
\Hy@unicodefalse	3119	\ifFld@multiline	2646
\Hy@VerboseAnchor	607	\ifFld@popdown	2704
\Hy@VerboseLinkStart	686, 709	\ifFld@radio	2700, 2723
\Hy@VerboseLinkStop		\ifHy@implicit	3521
	693, 729, 770, 837, 897, 941	\ifHy@pdfescapeform	3065, 3075
\Hy@VersionChecked	159	\ifHy@unicode	3118
\Hy@WrapperDef	603	\kv@set@family@handler	179
\Hy@xspace@end	690, 769, 836, 896, 940	\m@ne	3577, 3592
\HyAnn@AbsPageLabel		\OBJ@OCG@view	165
	2609, 2615, 2662,	\p@	2692
	2759, 2803, 2831, 2870, 2917, 2955	\pdf@ifdraftmode	2560
\HyAnn@Count	2613, 2614, 2617, 2618,	\PDF@SetupDoc	160
	2619, 2620, 2621, 2622, 2627, 2631	\PDFForm@Check	2956, 2988, 3190
\HyField@AddToFields	2515, 2670,	\PDFForm@List	2804, 2807, 3277
	2792, 2809, 2840, 2896, 2925, 2989	\PDFForm@Name	
\HyField@afields	2484		2656, 2717, 2820, 2946, 3104
\HyField@AfterAuxOpen	2486, 2512, 2545	\PDFForm@Push	2832, 2838, 3243
\HyField@AuxAddToCoFields		\PDFForm@Radio	2760, 2780, 3312
	2497, 2509, 2530	\PDFForm@Reset	2918, 2924, 3436
\HyField@AuxAddToFields		\PDFForm@Submit	2871, 2890, 3401
	2490, 2506, 2548	\PDFForm@Text	2663, 2668, 3351

