

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.95r, released 2022-08-24

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

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

Starting with L^AT_EX-2022-06-01 (or a suitable `latex-dev`) loading the testphase package is no longer required and this can be simplified to

```
\DocumentMetadata
{
  %% 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.

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

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

2 Bookmarks / outlines

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

3 “Metadata”

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

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

- `pdflang` is deprecated. Instead `\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'`.

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

5 PDF page size (mediabox)

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

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

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

The problem is to know which value to use (with the memoir class e.g. `\stockwidth` should be used instead of `\paperwidth`), and detecting this not a `hyperref` task. Instead the packages which change these values should also set the PDF page size. Also there are too many actors here: `color/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 `cmyk`. `rgb` is the default. It does *not* change the model of text colors. Be aware that while the PDF format allows `cmyk` (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 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.

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.

<code>pdfview (setup key)</code> <code>pdfstartview (setup key)</code> <code>pdfremotestartview (setup key)</code>	So with this driver the options <code>pdfview</code> , <code>pdfstartview</code> and <code>pdfremotestartview</code> take the following options: <ul style="list-style-type: none"> • <code>Fit</code>, <code>FitB</code>, <code>FitH</code>, <code>FitV</code>, <code>FitBH</code>, <code>FitBV</code> which can be followed by a positive integer (separated by a space) or the keyword <code>null</code>. The number can be gives as a <i><dimension expression></i> surrounded with <code>\hypercalcbp</code>. The driver redefines this command to use <code>\dim_to_decimal_in_bp:n</code>.
--	--

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

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

nesting (*setup key*) Typically PDF viewer can handle only GoToR annotations pointing to a PDF. So normally the default value `pdf` of this key should not be changed. This key is useless in PDF context. The boolean is only used in the code for anchors/destination where nesting doesn't make sense. It should not be changed.

pdfborder (*setup key*) This key set accept as value three numbers or three numbers and an array describing a dash pattern, examples are `0 0 1` or `0 0 1 [3 2]`. The first two numbers should

linkborder (*setup key*)

urlborder (*setup key*)

runborder (*setup key*)

menuborder (*setup key*)

according to the reference set round corners, but PDF viewer seem to ignore it. The third number is the line width of the border. Settings done with `pdfborderstyle` should take precedence.

<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 are stored di-		
<code>pdfmoddate</code> (<i>setup key</i>)	rectly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> . 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 set the language in <code>\DocumentMetadata</code> instead.		

File I

hyperref-generic driver implementation

```

1 <*package>
2 <@@=hyp>
3 \ProvidesFile{hggeneric-testphase.def}[2022-08-24 v0.95r %
4   generic Hyperref driver for the LaTeX PDF management testphase bundle]
5
6 \RequirePackage{etoolbox} %why?
7 \ExplSyntaxOn
8 \file_input:n {hyperref-colorschemes.def}
9 \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

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

```

At first a message for the testing of the resource management

```

12 \cs_if_exist:NTF \DocumentMetadata
13 {

```

```

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

```

The pdfversion should be set in \DocumentMetadata

```

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

```

A generic message for ignored keys.

```

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

```

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

```

57 \msg_new:nnn
58 { hyp }
59 { pdfa-no-push-button }
60 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
61
62 \msg_new:nnn

```

```

63 { hyp }
64 { pdfa-no-reset-button }
65 { PDF/A:~Reset~action~is~prohibited }

```

pdf/A message for not allowed Named actions

```

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

```

A message if the destination name is empty.

```

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

```

A message if the destination check fails

```

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

```

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

```

84 \msg_new:nnn
85 { hyp }
86 { ignore-deprecated-or-unknown-option-in-pdf-version }
87 {
88   Option~'~#1'~is~unknown~or~deprecated~in\\
89   pdf~version~#2.~Ignored.
90 }
91 \msg_new:nnn
92 { hyp }
93 { ignore-deprecated-or-unknown-value-in-pdf-version }
94 {
95   Value~'~#1'~is~unknown~or~deprecated~in\\
96   pdf~version~#2.~Ignored.
97 }
98 \msg_new:nnn
99 { hyp }
100 { replace-deprecated-or-unknown-value-in-pdf-version }
101 {
102   Value~'~#1'~is~unknown~or~deprecated~in\\
103   pdf~version~#2. Value~'~#3'~is used instead.
104 }

```

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:

```

105 \msg_new:nnn
106 { hyp }
107 { unknown-key }

```



```

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

```

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

```

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

```

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

```

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

```

2 Variants

```

155 \cs_generate_variant:Nn\pdf_destination:nn {nf}
156 \cs_generate_variant:Nn\pdf_object_ref:n {e}

```

3 Overwriting/providing commands from hyperref

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

```

157 \chardef\Hy@VersionChecked=1 %don't check the version!
158 %\cs_set_protected:Npn \PDF@SetupDoc{}
159 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

`\hypercalcbp`

```

160 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

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

```

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

```

`\Acrobatmenu` should use the new internal link command

```

172 \RenewDocumentCommand \Acrobatmenu { m m }
173 {
174   \hyper@linknamed {#1} {#2}
175 }

```

`\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!

```

176 % TODO should go at some time ...
177 % \kv@set@family@handler{Hyp}
178 % { \msg_warning:nnx {hyp}{unknown-key-to-Hyp}{#1} }
179 \cs_set_protected:Npn \hypersetup #1
180 {
181   \kvsetkeys{Hyp} {#1}
182   \keys_set:nn { hyp / setup }{ #1 }
183 }
184 % TODO for now unknown keys should only give warnings.

```

```

185 \keys_define:nn { hyp / setup }
186 {
187   unknown .code:n =
188   {
189     \msg_warning:nnxxx { hyp } { unknown-key }
190     { hyp~/~setup }{ \l_keys_key_str } { #1 }
191   }
192 }

```

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.

```

193 \keys_define:nn { hyp / setup }
194 {
195   next-anchor .code:n =
196   {
197     \AddToHookNext{__hyp/dest/make}
198     {\Hy@MakeCurrentHref{#1}}
199   }
200 }

```

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`)

```

201
202 \bool_new:N \l__hyp_href_url_encode_bool
203 \bool_new:N \l__hyp_href_url_ismap_bool
204 \tl_new:N \l__hyp_href_url_protocol_tl
205 \tl_new:N \l__hyp_href_pdf_destination_tl
206 \tl_new:N \l__hyp_href_pdf_page_tl
207 \tl_new:N \l__hyp_href_run_parameter_tl
208 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
209
210
211 \keys_define:nn { hyp / href }
212 {
213   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
214   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
215   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
216   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
217   ,pdfremotestartview .code:n =
218   {
219     \keys_set:nn { hyp / setup }
220     { pdfremotestartview = #1 }
221   }
222   ,page .code:n =
223   {
224     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
225     \tl_set:Nn \Hy@href@page {#1}
226   }
227   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
228   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
229   ,nextactionraw .code:n =

```

```

230 { %perhaps some safety match later, see hyperref code
231   \tl_if_empty:nTF {#1}
232   {
233     \pdfdict_remove:nn{l_hyp/annot/A}{Next}
234   }
235   {
236     \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
237     \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
238     \keys_set:nn {hyp / setup }{ pdfnewwindow = true}
239   }
240 }
241 ,afrelationship .code:n =
242 {
243   \pdfdict_put:nnx
244   { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
245 }
246
247 }
248
249 \keys_define:nn { hyp / setup }
250 {
251   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
252   ,href / urlencode .default:n = {true}
253   ,href / urlencode .initial:n = {false}
254   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
255   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
256   ,href / format .code:n = { \cs_set:Nn \__hyp_href_url_format:{#1} }
257 }
258
259 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
260
261 \DeclareRobustCommand*{\href}[1][ ]{%
262   \mode_leave_vertical:
263   \hook_use:n{cmd/href/before}
264   \group_begin:
265   \keys_set:nn { hyp / href } {#1}
266   \bool_if:NTF \l__hyp_href_url_encode_bool
267   {
268     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
269   }
270   {
271     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
272   }
273   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
274 }
275
276 \begingroup
277   \catcode'\$=6 %
278   \catcode'\#=12 %
279   \gdef\href@\$1{\expandafter\href@split$1##\}%
280   \gdef\href@split$1#$2#$3\\$4{%
281     \hyper@@link{$1}{$2}{$4}%<---__hyp-docstrip doubling!
282   }
283   \hook_use:n{cmd/href/after}

```

```

284 }%
285 \endgroup
286
287 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
288
289 \DeclareRobustCommand*{\hrefurl}[1] []
290 {
291   \mode_leave_vertical:
292   \hook_use:n{cmd/href/before}
293   \group_begin:
294   \keys_set:nn { hyp / href } {#1}
295   \bool_if:NTF \l__hyp_href_url_encode_bool
296   {
297     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
298   }
299   {
300     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
301   }
302   \hyper@normalise\__hyp_href_url_aux:nn}
303
304 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
305 {
306   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
307   \group_end:
308   \hook_use:n{cmd/href/after}
309 }
310
311 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
312 \DeclareRobustCommand*{\hrefpdf}[1] []
313 {
314   \mode_leave_vertical:
315   \hook_use:n{cmd/hrefpdf/before}
316   \group_begin:
317   \keys_set:nn { hyp / href } {#1}
318   \hyper@normalise\__hyp_href_pdf_aux:nn
319 }
320
321 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
322 {
323   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
324   \group_end:
325   \hook_use:n{cmd/hrefpdf/after}
326 }
327
328 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
329 \DeclareRobustCommand*{\hrefrun}[1] []
330 {
331   \mode_leave_vertical:
332   \hook_use:n{cmd/hrefrun/before}
333   \group_begin:
334   \keys_set:nn { hyp / href } {#1}
335   \hyper@normalise\__hyp_href_run_aux:nn
336 }
337

```

```

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

```

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

```

373
374 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
375   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
376   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
377   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
378   #2}
379

```

make the new commands compatible with `\pdfstringdef`:

```

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

```
384 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
385 {
386   %\tl_set:cn {@#1}{#2}
387   \AddToDocumentProperties[hyperref]{#1}{#2}
388 }
389 \cs_generate_variant:Nn \__hyp_store_metadata:nn {xn,nx,xx}
```

4.2 citecolor

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

```
390 \hook_new:n{hyp/link/cite}
391 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
392 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
393 \keys_define:nn { hyp / setup }
394 {
395   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init
396   ,citebordercolor
397     .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_i
398 }
399 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
400 {
401   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
402   {
403     \keys_set:nn { hyp / setup }
404     {
405       linkcolor      = hyp/color/cite
406     }
407   }
408   \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
409 }
410 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
411 {
412   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
413   {
414     \keys_set:nn { hyp / setup }
415     {
416       linkbordercolor = hyp/color/citeborder
417     }
418   }
419   \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
420 }
421
```

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

```

422 \bool_lazy_and:nnF
423 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
424 { \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

```

425 \AddToHook { package/bookmark/after}
426 {
427   \define@key{BKM}{color}
428   {
429     \__hyp_color_set:ne {__hyp/tmpa}{#1}
430     \color_export:nVN
431       {__hyp/tmpa}
432       \g__hyp_bordercolormodel_str
433       \BKM@color
434   }
435 }
436 \legacy_if:nT { Hy@bookmarks }
437 {
438   \AddToHook{begindocument/before}[hyperref/bookmark]
439   {
440     \RequirePackage{bookmark}
441   }
442 }
443 \legacy_if:nT { Hy@draft}
444 {
445   \PassOptionsToPackage{draft}{bookmark}
446 }

```

6 Reference and label commands

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

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

```

\__hyp_ref_label:nn
\__hyp_ref_if_exist:nn 447 %
\__hyp_ref_check:nn 448 \cs_new_protected:Npn \__hyp_ref_label:nn #1 #2 %label/attributes
\__hyp_ref_value:nn 449 {
450   \@bsphack
451   \ref_label:nn{#1}{#2}
452   \@esphack
453 }
454 \cs_generate_variant:Nn \__hyp_ref_label:nn {en}

```


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

```

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

```

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

```

457 \cs_new_protected:Npn \__hyp_ref_check:nn #1 #2 %label/attribute
458 {
459   \__hyp_ref_if_exist:nnF {#1}{#2}
460   {
461     \protect\G@refundefinedtrue
462     \@latex@warning
463     {
464       Reference~'\tl_to_str:n {#1}'~with~attribute~'\tl_to_str:n {#2}'~
465       on~page~\thepage~\space undefined
466     }
467   }
468 }
469 \cs_generate_variant:Nn \__hyp_ref_check:nn {en}

```

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

```

470 \cs_new:Npn \__hyp_ref_value:nn #1 #2
471 {
472   \ref_value:nn{#1}{#2}
473 }
474 \cs_generate_variant:Nn \__hyp_ref_value:nn {en}

```

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

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpa_seq 475 \box_new:N \l__hyp_tmpa_box
\l__hyp_tmpa_int 476 \tl_new:N \l__hyp_tmpa_tl
\l__hyp_tmpa_box 477 \seq_new:N \l__hyp_tmpa_seq
\l__hyp_tmpa_str 478 \int_new:N \l__hyp_tmpa_int
479 \str_new:N \l__hyp_tmpa_str

```

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

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

TODO: document and check use!

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl 480 \tl_new:N \l__hyp_dest_name_tmpa_tl
\l__hyp_filename_tmpa_tl 481 \tl_new:N \l__hyp_uri_tmpa_tl
\__hyp_text_tmpa_str_uu\g__hyp_text_tmpa_str 482 \tl_new:N \l__hyp_filename_tmpa_tl
483 \tl_new:N \l__hyp_para_tmpa_tl
484 \str_new:N \l__hyp_text_tmpa_str
485 \str_new:N \g__hyp_text_tmpa_str

```

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

7.2 Constants

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

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

(End definition for `\c__hyp_dest_undefined_tl`.)

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

```
487 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
488 {url,link,file,menu,run}
489 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
490 {
491   URI    = url,
492   GoTo   = link,
493   GoToR  = file,
494   Named  = menu,
495   Launch = run
496 }
497 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
498 {
499   url    = URI,
500   link   = GoTo,
501   file   = GoToR,
502   menu   = Named,
503   run    = Launch
504 }
505
```

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

7.3 Variables

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

```
506 \tl_new:N \g__hyp_dest_pdfstartpage_tl
507 \tl_new:N \g__hyp_dest_pdfstartview_tl
508 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

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

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

```
\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
509 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
510 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
511 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
512 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
513 \tl_new:N \l__hyp_text_enc_file_print_tl
514 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

515
516 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
517 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
518 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
519 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
520 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
521 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

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

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

```

522 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

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

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

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

```

523 \str_new:N \g__hyp_bordercolormodel_str

```

(End definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorrun_bool`

`\l_hyp_annot_colormenu_bool`

```

524 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

525 {

```

```

526   \bool_new:c {l_hyp_annot_color#1_bool}

```

```

527 }

```

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

`\l_hyp_annot_ocgcolorlink_bool`

`\l_hyp_annot_ocgcolorurl_bool`

`\l_hyp_annot_ocgcolorfile_bool`

`\l_hyp_annot_ocgcolorrun_bool`

`\l_hyp_annot_ocgcolormenu_bool`

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

```

528 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

529 {

```

```

530   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}

```

```

531 }

```

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

`\l_hyp_annot_Named_bool`

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

```

532 \seq_map_inline:Nn \c_pdfannot_link_types_seq

```

```

533 {

```

```

534   \bool_new:c {l__hyp_annot_#1_bool}

```

```

535   \bool_set_true:c {l__hyp_annot_#1_bool}

```

```

536 }

```

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

7.5 Boxes

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

```
537 \box_new:N \l__hyp_dest_box
```

(End definition for \l__hyp_dest_box.)

7.6 Regex

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

```
538 \regex_const:Nn \c__hyp_dest_startview_regex
539 {
540   \A\ *
541   (?:
542     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
543     |
544     (?:Fit\b|FitB\b)
545     |
546     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
547     |
548     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
549   )
550 }
```

(End definition for \c__hyp_dest_startview_regex.)

7.7 PDF dictionaries

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

```
551 \pdfdict_new:n {l__hyp_page/Trans}
552 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End definition for l__hyp_page/Trans.)

8 PDF string conversion

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

`hyp/text/pdfstring`

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

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

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

`__hyp_text_purify:nN`

```

554 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
555 {
556   \str_set:Nx #2 {\text_purify:n { #1 } }
557 }

```

(End definition for __hyp_text_purify:nN.)

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

`__hyp_text_cleanup:N`

```

558 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
559 {
560
561 }

```

(End definition for __hyp_text_cleanup:N.)

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

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

utf8/string	(lit) (utf8/string)
utf8/string-raw	lit (utf8/string)
utf8/URI	(percent encoded url)
utf8/URI-raw	percent encoded url
utf16/hex	<HEX> (utf16/hex)
utf16/hex-raw	HEX (utf16/hex)
utf16/string	(lit) (utf16/string)
utf16/string-raw	lit (utf16/string)

`__hyp_text_string_from_unicode:nN`

```

562 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
563 {
564   \pdf_string_from_unicode:nVN { #1 } #2 #2
565 }

```

(End definition for __hyp_text_string_from_unicode:nN.)

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

`__hyp_text_pdfstring:nnN`

```

566 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
567 {
568   \group_begin:
569   \Hy@pdfstringtrue
570   \hook_use:n {hyp/text/pdfstring}
571   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
572   \__hyp_text_cleanup:N \l__hyp_text_tmpa_str
573   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
574   \str_gset_eq:NN \g__hyp_text_tmpa_str \l__hyp_text_tmpa_str
575   \group_end:
576   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
577 }
578 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {xnN,onN,xoN,ooN,noN}

```

```
(End definition for \_hyp\_text\_pdfstring:nnN.)
!!! temporary until all instances are gone
579 \cs_new_protected:Npn \Hy@pstringdef #1 #2
580 { \_hyp\_text\_pdfstring:xnN {#2} {utf8/string-raw}#1 }
This is a special version for info keys:
```

```
\_hyp\_text\_pdfstring\_info:nnN
```

```
581 \cs_new_protected:Npn \_hyp\_text\_pdfstring\_info:nnN #1 #2
582 {
583   \_hyp\_text\_pdfstring:nonN { #1 } { \l\_hyp\_text\_enc\_info\_print\_tl } #2
584 }
```

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

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
```

```
585 \cs_new_protected:Npn \_hyp\_PageLabels\_gpush:
586 {
587   \pdfmanagement\_add:nnx {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
```

```

588 }
589
590 \def\Hy@PutCatalog #1 {}
591
592
593 \legacy_if:nT { Hy@pdfpagelabels }
594 {
595   \cs_set_protected:Npn \HyPL@StorePageLabel #1
596   {
597     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
598     \__hyp_PageLabels_gpsh:
599   }
600 }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn

601 \Hy@WrapperDef \__hyp_destination:nn #1 #2
602 {
603   \mode_if_horizontal:T { \@savsf\spacefactor }
604   \Hy@SaveLastskip      %defined in hyperref
605   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
606   \__hyp_text_pdfstring:xoN
607   { \HyperDestNameFilter{#1} }

```

```

608     { \l__hyp_text_enc_dest_tl }
609     \l__hyp_tmpa_tl
610 \str_if_eq:nnTF {#2} {fitrbox}
611 {
612     \exp_args:NV
613     \pdf_destination:nnnn \l__hyp_tmpa_tl
614     { \box_wd:N \l__hyp_dest_box }
615     { \box_ht:N \l__hyp_dest_box }
616     { \box_dp:N \l__hyp_dest_box }
617 }
618 {
619     \exp_args:NV
620     \pdf_destination:nf
621     { \l__hyp_tmpa_tl }
622     { #2 }
623 }
624 \Hy@RestoreLastskip %defined in hyperref
625 \mode_if_horizontal:T { \spacefactor\@savsf }
626 }

```

(End definition for `__hyp_destination:nn`.)

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

```

\hyper@anchor
\hyper@anchorstart 627 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 628 \hook_new:n{hyp/anchor}
hyp/anchor 629 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 630 {
631     \exp_args:NnV
632     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
633     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
634     \hook_use:n{hyp/anchor}
635 }
636
637 \cs_new_protected:Npn \hyper@anchorstart #1
638 {
639     \Hy@activeanchortrue
640     \exp_args:NnV
641     \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
642     \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
643     \hook_use:n{hyp/anchor}
644 }
645
646 \cs_new_protected:Npn \hyper@anchorend
647 {
648     \Hy@activeanchorfalse
649 }

```

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

10.2 GoTo Links

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

At first the internal link commands:

```

650 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
651 {
652   \mode_leave_vertical:
653   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
654   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
655   {
656     \msg_warning:nnx
657     { hyp }
658     { empty-destination-name }
659     { \c__hyp_dest_undefined_tl }
660     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
661   }
662   {
663     \__hyp_text_pdfstring:xoN
664     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
665     { \l__hyp_text_enc_dest_tl }
666     \l__hyp_dest_name_tmpa_tl
667   }
668   \exp_args:No
669   \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
670 }
671
672 \cs_new_protected:Npn \__hyp_link_goto_end:
673 {
674   \pdfannot_link_goto_end:
675 }
```

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

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

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

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

```

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

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

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

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

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

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

```

676
677 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
678 {
679   \bool_if:NTF \l__hyp_annot_GoTo_bool
680   {
681     \Hy@VerboseLinkStart{#1}{#2}
682     \group_begin:
683     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
684     \hook_use:n {hyp/link/#1}
685     \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
686     \__hyp_link_goto_end:
687     \group_end:
688     \Hy@VerboseLinkStop
689   }
690   {\let\protect\relax#3}}
691 }
692
693 \cs_new_protected:Npn \hyper@linkstart #1 #2 % #1 context, #2=destination name
694 {
695   \bool_if:NT \l__hyp_annot_GoTo_bool
696   {
697     \Hy@VerboseLinkStart{#1}{#2}% only for debug
698     \group_begin:
699     \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
700     \hook_use:n {hyp/link/#1}
701     \__hyp_link_goto_begin:nw {#2}
702   }
703 }
704
705 \cs_new_protected:Npn \hyper@linkend
706 {
707   \bool_if:NT \l__hyp_annot_GoTo_bool
708   {
709     \__hyp_link_goto_end:
710     \group_end:
711     \Hy@VerboseLinkStop
712   }
713 }
```

10.3 URI links

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

```

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

```

715 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
716 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
717
718 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
719 {
720   \bool_if:NTF \l__hyp_annot_URI_bool
721   {
722     \group_begin:
723     \__hyp_text_pdfstring:xoN
724     { #2}
725     { \l__hyp_text_enc_uri_print_tl }
726     \l__hyp_uri_tmpa_tl
727     \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
728     \bool_if:NT \l__hyp_href_url_ismap_bool
729     {
730       \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
731     }
732     \cs_set_eq:NN \# \c_hash_str
733     \cs_set_eq:NN \% \c_percent_str
734     \Hy@safe@activetrue
735     \mode_leave_vertical:
736     \pdfannot_dict_put:nxx {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
737     \pdfannot_link:nxx { URI }
738     {
739     }
740     {
741       \let\protect\relax
742       #1
743       \Hy@xspace@end
744       \Hy@VerboseLinkStop %where is the start??
745     }
746     \group_end:
747   }
748   {{\let\protect\relax#1}}
749 }
750

```

10.4 GoToR Links files

```

751 \pdfdict_new:n {l_hyp/annot/A/GoToR}
752 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
753 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
754
755
756 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
757 {
758   \bool_if:NTF \l__hyp_annot_GoToR_bool
759   {
760     \group_begin:
761     \tl_set:Nx \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
762     \exp_args:Nx
763     \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
764     {
765       \pdfdict_put:nxx { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n {ap

```

```

766         \exp_args:Nnox
767         \pdffile_embed_file:nnn
768         {}
769         {\l__hyp_filename_tmpa_tl }
770         {\__hyp_file_tl_to_str:N \l__hyp_filename_tmpa_tl }
771     }
772     \pdfdict_put:nnx
773     {l_hyp/annot/A/GoToR}
774     {F}
775     {\pdf_object_ref:e {\__hyp_file_tl_to_str:N \l__hyp_filename_tmpa_tl}}
776     \__hyp_text_pdfstring:nnN
777     { #3 }
778     { \l__hyp_text_enc_dest_print_tl }
779     \l__hyp_dest_name_tmpa_tl
780     \tl_if_blank:eTF {#3}
781     {
782         \pdfdict_put:nnx {l_hyp/annot/A/GoToR}{D}
783         {
784             [
785                 \int_eval:n
786                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
787                 /\l__hyp_dest_pdfremotestartview_tl
788             ]
789         }
790     }
791     {
792         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
793     }
794     \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>

```

795     \pdf_object_unnamed_write:nx{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
796     \pdfannot_dict_put:nnx {link/GoToR}{A}{\pdf_object_ref_last:}
797     \pdfannot_link:nnx %expansion??
798     { GoToR }
799     {
800     }
801     {
802         \let\protect\relax
803         #1\Hy@xspace@end
804         \Hy@VerboseLinkStop %where is the start??
805     }
806     \group_end:
807 }
808 {{\let\protect\relax#1}}
809 }

```

10.5 Launch links

We define \hyper@linklaunch for naming consistency

```

810 \pdfdict_new:n {l_hyp/annot/A/Launch}
811 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
812 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
813

```

```

814 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
815 {
816   \bool_if:NTF \l__hyp_annot_Launch_bool
817   {
818     \group_begin:
819     \__hyp_text_pdfstring:nnN
820       { #1 }
821       { \l__hyp_text_enc_file_print_tl }
822       \l__hyp_filename_tmpa_tl
823     \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
824     \__hyp_text_pdfstring:noN
825       { #3 }
826       { \l__hyp_text_enc_para_print_tl }
827       \l__hyp_para_tmpa_tl
828     \bool_if:nTF
829     {
830       \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}
831       ||
832       \pdf_version_compare_p:Nn > {1.9}
833     }
834     {
835       \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
836     }
837     {
838       \pdfdict_put:nnx {l_hyp/annot/A/Launch}{Win}
839       {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
840     }
841     \mode_leave_vertical:
842     \pdfannot_dict_put:nnx {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}>>}
843     \pdfannot_link:nxn
844       { Launch }
845       {
846         % /A
847         % <<
848         % \pdfdict_use:n {l_hyp/annot/A/Launch}
849         % >>
850       }
851     {
852       \let\protect\relax
853       #2\Hy@xspace@end
854       \Hy@VerboseLinkStop %where is the start??
855     }
856     \group_end:
857   }
858   {\let\protect\relax#2}}
859 }

```

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

```

860 \use:x
861 { % filename, anchor text, linkname
862   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str ##1 \exp_not:N \ \ ##2 #
863 }
864 {

```

```

865 \hyper@linklaunch {#1}{#2}{#3}
866 }

```

10.6 Named links (menu)

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

```

867 \pdfdict_new:n {l_hyp/annot/A/Named}
868 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
869 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
870
871 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
872 {
873   \bool_if:NTF \l__hyp_annot_Named_bool
874   {
875     \group_begin:
876     \pdfmeta_standard_verify:nnTF {named_actions}{#1}
877     {
878       \mode_leave_vertical:
879       \pdfdict_put:nnx {l_hyp/annot/A/Named}{N}
880       {\pdf_name_from_unicode_e:n{#1}}
881       \pdfannot_dict_put:nnx {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}>>}
882       \pdfannot_link:nnx { Named }
883       {
884         % /A
885         % <<
886         % \pdfdict_use:n { l_hyp/annot/A/Named }
887         % >>
888       }
889       {
890         #2
891         \Hy@xspace@end
892         \Hy@VerboseLinkStop
893       }
894     }
895     {
896       \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
897       #2
898     }
899     \group_end:
900   }
901   {{\let\protect\relax#2}}
902 }
903

```

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.

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

_hyp_color_select:n _hyp_color_select:n {<color>}

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

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

```

905 \cs_new_protected:Npn \_hyp_color_select:n #1
906 {
907   \tl_if_head_eq_charcode:nNTF {#1}[ %]
908   {
909     \_hyp_color_select_aux:wn #1
910   }
911   {
912     \color_select:n {#1}
913   }
914 }
915
916 \cs_new_protected:Npn \_hyp_color_select_aux:wn [#1] #2
917 {
918   \color_select:nn {#1}{#2}
919 }
920
921 \cs_generate_variant:Nn \_hyp_color_select:n {e}

```

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

_hyp_color_set:nn _hyp_color_set:nn {< name >} {<color>}

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

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

```

922 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
923 {
924   \tl_if_head_eq_charcode:nNTF {#2}[ %]
925   {
926     \_hyp_color_set_aux:nwn { #1 } #2
927   }
928   {
929     \color_set:nn {#1} {#2}
930   }
931 }
932
933 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
934 {
935   \color_set:nnn {#1}{#2}{#3}
936 }
937
938 \cs_generate_variant:Nn \_hyp_color_set:nn {ne}

```

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

11.2 Textcolor of links

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

```

939 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
940 {
941   \hook_gput_code:nnn
942     {pdfannot/link/#2/begin}
943     {hyp/color}
944   {
945     \bool_if:cT { l_hyp_annot_color#1_bool }
946     {
947       \group_begin:
948       \color_select:n { hyp/color/#1}
949     }
950   }
951   \hook_gput_code:nnn
952     {pdfannot/link/#2/end}
953     {hyp/color}
954   {
955     \bool_if:cT { l_hyp_annot_color#1_bool }
956     {
957       \group_end:
958     }
959   }
960 }
```

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

```

961 \keys_define:nn { hyp / setup }
962 {
963   ,colorlinks .meta:n =
964   {
965     ,pdfborder={0~0~0}
966     ,pdfborderstyle=
967     ,colorurl =#1
968     ,colorlink =#1
969     ,colorrurl =#1
970     ,colormenu =#1
971     ,colorfile =#1
972   }
973   ,colorlinks .default:n = {true}
974 }
```

`colorurl` (*setup key*)

`colorlink` (*setup key*)

`colorrurl` (*setup key*)

`colormenu` (*setup key*)

`colorfile` (*setup key*)

`urlcolor` (*setup key*)

`linkcolor` (*setup key*)

`runcolor` (*setup key*)

`menucolor` (*setup key*)

`filecolor` (*setup key*)

`allcolors` (*setup key*)

```

975 \seq_map_inline:Nn \c__hyp_annot_types_seq
976 {
977   \keys_define:nn { hyp / setup }
978   {
979     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
980     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
981   }
982 }
983
```



```

984 \keys_define:nn { hyp / setup }
985 {
986   ,allcolors .meta:n =
987   {
988     ,urlcolor=#1
989     ,linkcolor=#1
990     ,runcolor=#1
991     ,filecolor=#1
992     ,menucolor=#1
993   }
994   ,allcolors .value_required:n = true
995 }

```

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*)

```

996 \keys_define:nn { hyp / setup }
997 {
998   ,bordercolormodel .choices:nn =
999   {rgb,cmyk}
1000   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1001   ,bordercolormodel .initial:n = {rgb}
1002 }

1003 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1004 {
1005   \keys_define:nn { hyp / setup }
1006   {
1007     #1bordercolor .code:n =
1008     {
1009       \tl_if_empty:nTF { ##1 }
1010       {
1011         \pdfannot_dict_remove:nn
1012         {link/#2}
1013         { C }
1014       }
1015       {
1016         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1017         \color_export:nVN
1018         {hyp/color/#1border}
1019         \g__hyp_bordercolormodel_str
1020         \l__hyp_tmpa_tl
1021         \pdfannot_dict_put:nnx
1022         {link/#2}
1023         { C }
1024         { [\l__hyp_tmpa_tl] }
1025       }
1026     }
1027   }
1028 }

```

```

1029
1030 \keys_define:nn { hyp / setup }
1031 {
1032   ,allbordercolors .meta:n =
1033   {
1034     ,linkbordercolor=#1
1035     ,urlbordercolor =#1
1036     ,filebordercolor=#1
1037     ,menubordercolor=#1
1038     ,runbordercolor =#1
1039   }
1040   ,allbordercolors .value_required:n = true
1041 }
1042

```

11.3.2 Borderwidth and -arc

```

1043 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1044 {
1045   \keys_define:nn { hyp / setup }
1046   {
1047     #1border .code:n =
1048     {
1049       \tl_if_empty:nTF { ##1 }
1050       {
1051         \pdfannot_dict_remove:nn
1052         {link/#2}
1053         { Border }
1054       }
1055       {
1056         \pdfannot_dict_put:nnn
1057         {link/#2}
1058         { Border }
1059         { [##1] }
1060       }
1061     }
1062   }
1063 }
1064 \keys_define:nn { hyp / setup }
1065 {
1066   ,pdfborder .code:n =
1067   {
1068     \tl_if_empty:nTF { #1 }
1069     {
1070       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1071       {
1072         \pdfannot_dict_remove:nn
1073         {link/##2}
1074         { Border }
1075       }
1076     }
1077     {
1078       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1079       {

```

```

1080         \pdfannot_dict_put:nnn
1081         {link/##2}
1082         { Border }
1083         { [#1] }
1084     }
1085 }
1086 }
1087 ,pdfborder .initial:n = {0~0~1},
1088 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1089 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1090 {
runborderstyle (setup key) 1091     \keys_define:nn { hyp / setup }
fileborderstyle (setup key) 1092     {
menuborderstyle (setup key) 1093         #1borderstyle .code:n =
1094         {
1095             \tl_if_empty:nTF { ##1 }
1096             {
1097                 \pdfannot_dict_remove:nn
1098                 {link/#2}
1099                 { BS }
1100             }
1101             {
1102                 \pdfannot_dict_put:nnn
1103                 {link/#2}
1104                 { BS }
1105                 { <<##1>> }
1106             }
1107         }
1108     }
1109 }
1110 \keys_define:nn { hyp / setup }
1111 {
1112     ,pdfborderstyle .code:n =
1113     {
1114         \tl_if_empty:nTF { #1 }
1115         {
1116             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1117             {
1118                 \pdfannot_dict_remove:nn
1119                 {link/##2}
1120                 { BS }
1121             }
1122         }
1123         {
1124             \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1125             {
1126                 \pdfannot_dict_put:nnn
1127                 {link/##2}
1128                 { BS }

```

```

1129         { <<#1>> }
1130     }
1131 }
1132 }
1133 ,pdfborderstyle .initial:n = {},
1134 }

```

11.4 ocolorlinks

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.

```

1135 \cs_new_protected:Npn \_hyp_ocg_init:
1136 {
1137     \pdf_object_new:nn { __hyp/OCG/View } { dict }
1138     \pdf_object_new:nn { __hyp/OCG/Print } { dict }
1139     \pdf_object_new:nn { __hyp/OCG/config } { dict }
1140     \pdf_object_new:nn { __hyp/OCG/refarray } { array }
1141     \pdf_object_write:nx { __hyp/OCG/refarray }
1142     {
1143         \pdf_object_ref:n { __hyp/OCG/View }
1144         \c_space_tl
1145         \pdf_object_ref:n { __hyp/OCG/Print }
1146     }
1147     \pdf_object_write:nn { __hyp/OCG/View }
1148     {
1149         /Type/OCG
1150         /Name(View)
1151         /Usage
1152         <<
1153         /Print <</PrintState/OFF>>~
1154         /View <</ViewState/ON >>~
1155         >>
1156     }
1157     \pdf_object_write:nn { __hyp/OCG/Print }
1158     {
1159         /Type/OCG
1160         /Name(Print)
1161         /Usage
1162         <<
1163         /Print <</PrintState/ON>>~
1164         /View <</ViewState/OFF>>~
1165         >>
1166     }
1167     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/View} }
1168     \pdfmanagement_add:nnx { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OCG/Print} }
1169     \pdf_object_write:nx { __hyp/OCG/config }
1170     {
1171         /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1172         /AS[
1173         <<
1174         /Event/View

```

```

1175         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1176         /Category[/View]
1177     >>
1178     <<
1179         /Event/Print
1180         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1181         /Category[/Print]
1182     >>
1183     <<
1184         /Event/Export
1185         /OCGs\c_space_t1 \pdf_object_ref:n { __hyp/OCG/refarray }
1186         /Category[/Print]
1187     >>
1188     ]
1189 }
1190 \pdfmanagement_add:nnx { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1191 \cs_gset:Npn \__hyp_ocg_init: {}
1192 }

```

(End definition for __hyp_ocg_init:.)

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

```

1193 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1194 {
1195     \hook_gput_code:nnn
1196     {pdfannot/link/#2/begin}
1197     {hyp/ocg}
1198     {
1199         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1200         {
1201             \__hyp_ocg_init:
1202             \group_begin:
1203             \hbox_set:Nw \l__hyp_tmpa_box
1204         }
1205     }
1206     \hook_gput_code:nnn
1207     {pdfannot/link/#2/end}
1208     {hyp/ocg}
1209     {
1210         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1211         {
1212             \hbox_set_end:
1213             \mbox
1214             {
1215                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1216                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1217                 \pdf_emc:
1218                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1219                 \group_begin:
1220                 \color_select:n { hyp/color/#1 }
1221                 \box_use_drop:N \l__hyp_tmpa_box
1222                 \group_end:
1223                 \pdf_emc:
1224             }

```

```

1225         \group_end:
1226     }
1227 }
1228 }

```

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

```

1229 \bool_lazy_or:nnTF
1230 { \pdf_version_compare_p:Nn > {1.4} }
1231 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1232 {
1233     \keys_define:nn { hyp / setup }
1234     {
1235         ,_ocgcolorlinks .meta:n =
1236         {
1237             ocgcolorlink=#1,
1238             ocgcolorurl=#1,
1239             ocgcolorfile=#1,
1240             ocgcolorrun=#1,
1241             ocgcolormenu=#1
1242         }
1243         ,_ocgcolorlinks .default:n = true
1244     }
1245 }
1246 {
1247     \keys_define:nn { hyp / setup }
1248     {
1249         ,_ocgcolorlinks .code:n =
1250         {
1251             \msg_warning:nnxx
1252             { hyp }
1253             { ignore-deprecated-or-unknown-option-in-pdf-version }
1254             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1255         }
1256     }
1257 }
1258
1259 \keys_define:nn { hyp / setup }
1260 {
1261     ,ocgcolorlinks .choice:
1262     ,ocgcolorlinks / true .meta:n =
1263     {
1264         pdfborder      = {0~0~0},
1265         pdfborderstyle = {},
1266         colorlinks     = false,
1267         _ocgcolorlinks = true
1268     }
1269     ,ocgcolorlinks / false .meta:n =
1270     {
1271         _ocgcolorlinks = false
1272     }
1273     ,ocgcolorlinks .default:n = {true}
1274 }
1275
1276 \seq_map_inline:Nn \c__hyp_annot_types_seq

```

```

1277 {
1278   \bool_lazy_or:nnTF
1279   { \pdf_version_compare_p:Nn > {1.4} }
1280   { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1281   {
1282     \keys_define:nn { hyp / setup }
1283     {
1284       ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1285     }
1286   }
1287   {
1288     \keys_define:nn { hyp / setup }
1289     {
1290       ,ocgcolor#1 .code:n=
1291       {
1292         \msg_warning:nnxx
1293         { hyp }
1294         { ignore-deprecated-or-unknown-option-in-pdf-version }
1295         { ocgcolor#1 }
1296         { \pdf_version_major:.\pdf_version_minor: }
1297       }
1298     }
1299   }
1300 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1301 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1302 {
1303   \keys_define:nn { hyp / setup }
1304   {
1305     ,#1highlight .choices:nn =
1306     { /I, /N, /O, /P }
1307     {
1308       \pdfannot_dict_put:nnn
1309       {link/#2}
1310       { H }
1311       { ##1 }
1312     }
1313     ,#1highlight / .code:n =
1314     {
1315       \pdfannot_dict_remove:nn
1316       {link/#2}
1317       { H }
1318     }
1319   }
1320   ,#1highlight / unknown .code:n =
1321   {
1322     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1323     { pdfhighlight }
1324     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1325     { \exp_not:n {##1} }
1326   }

```

```

1327     }
1328   }
1329 }
1330
1331
1332 \keys_define:nn { hyp / setup }
1333 {
1334   ,pdfhighlight .choices:nn =
1335   { /I, /N, /O, /P}
1336   {
1337     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1338     {
1339       \pdfannot_dict_put:nnn
1340       {link/####2}
1341       { H }
1342       { #1 }
1343     }
1344   }
1345   ,pdfhighlight / .code:n =
1346   {
1347     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1348     {
1349       \pdfannot_dict_remove:nn
1350       {link/##2}
1351       { H }
1352     }
1353   }
1354   ,pdfhighlight .initial:n = {/I},
1355   ,pdfhighlight / unknown .code:n =
1356   {
1357     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1358     { pdfhighlight }
1359     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1360     { \exp_not:n {#1} }
1361   }
1362 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1363 \keys_define:nn { hyp / setup }
hideurl (setup key) 1364 {
hidefile (setup key) 1365   hidelinks .meta:n =
hiderun (setup key) 1366   {
hidemenu (setup key) 1367     ,colorlinks      = false
1368     ,ocgcolorlinks = false
1369     ,pdfborder      = { 0~0~0 }
1370     ,pdfborderstyle=
1371   }
1372 }
1373
1374 \seq_map_inline:Nn \c__hyp_annot_types_seq

```



```

1375 {
1376   \keys_define:nn { hyp / setup }
1377   {
1378     hide#1 .meta:n =
1379     {
1380       ,color#1      = false
1381       ,ocgcolor#1   = false
1382       ,#1border     = { 0~0~0 }
1383       ,#1borderstyle =
1384     }
1385   }
1386 }

```

11.7 color schemes and settings

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

`colorscheme (setup key)`

```

1387 \keys_define:nn { hyp / setup }
1388 {
1389   colorscheme .code:n =
1390   {
1391     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1392     {
1393       \keys_set:nn { hyp / setup }
1394       {
1395         ##1 = ##2
1396       }
1397     }
1398   }
1399 }
1400 \keys_set:nn { hyp / setup } {colorscheme=phetype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

`unicode (setup key)`

`pdfencoding (setup key)`

`pdfversion (setup key)`

```

1401 \keys_define:nn { hyp / setup }
1402 {
1403   ,unicode      .code:n = {}
1404   ,pdfencoding .code:n = {}
1405   ,pdfversion .code:n =
1406   {
1407     \msg_warning:nn { hyp }{ pdfversion-disabled }
1408   }
1409 }
1410 %

```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```

verbose (setup key)
  debug (setup key) 1411 \keys_define:nn { hyp / setup }
  draft (setup key) 1412 {
  final (setup key) 1413   ,verbose .legacy_if_set:n = {Hy@verbose}
                  1414   ,debug .legacy_if_set:n = {Hy@verbose}
                  1415 }
                  1416 \keys_define:nn { hyp / setup }
                  1417 {
                  1418   ,draft .code:n =
                  1419   {
                  1420     \Hy@drafttrue
                  1421     \PassOptionsToPackage{draft}{bookmark}
                  1422   }
                  1423   ,final .code:n =
                  1424   {
                  1425     \Hy@finaltrue
                  1426     \PassOptionsToPackage{final}{bookmark}
                  1427   }
                  1428 }

extension (setup key)
hypertextnames (setup key) 1429 \keys_define:nn { hyp / setup }
naturalnames (setup key) 1430 {
  pageanchor (setup key) 1431   ,extension .tl_set:N = \XR@ext
  linktoc (setup key) 1432   ,extension .initial:n= pdf
  linktocpage (setup key) 1433   ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
  plainpages (setup key) 1434   ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1435   ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1436   ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
                  1437   ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
                  1438   ,plainpages .legacy_if_set:n = {Hy@plainpages}
                  1439 }
                  1440
                  1441 \keys_define:nn { hyp / setup }
                  1442 {
                  1443   ,linktoc .choices:nn = { none, section, all, page }
                  1444   {
                  1445     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
                  1446   }
                  1447   ,linktoc / unknown .code:n =
                  1448   {
                  1449     \msg_warning:nnxxx { hyp } { unknown-choice }
                  1450     { linktoc }
                  1451     { none, section, all, page }
                  1452     { \exp_not:n {#1} }
                  1453   }
                  1454   ,linktocpage .choice:
                  1455   ,linktocpage / true .meta:n = {linktoc=page}
                  1456   ,linktocpage / false .meta:n = {linktoc=section}
                  1457   ,linktocpage .default:n = true

```

```

1458 }
1459
link (setup key) This booleans allow to disable the link types.
url (setup key) 1460 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1461 {
menu (setup key) 1462 \keys_define:nn { hyp / setup }
run (setup key) 1463 {
1464 ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1465 }
1466 }

1467 \keys_define:nn { hyp / setup }
1468 {
1469 ,baseurl .code:n =
1470 {
1471 \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1472 \tl_if_empty:NTF \l__hyp_tmpa_tl
1473 {
1474 \pdfmanagement_remove:nn {Catalog} { URI }
1475 }
1476 {
1477 \pdfmanagement_add:nxx {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1478 }
1479 }
1480 %only false does something ...
1481 ,bookmarks .choice:
1482 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1483 ,bookmarks / true .code:n = {}
1484 ,bookmarks .default:n = {true}
1485 ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1486 ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1487 ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1488 ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1489 ,pdfcenterwindow .choice:
1490 ,pdfcenterwindow / false .code:n =
1491 {
1492 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1493 }
1494 ,pdfcenterwindow / true .code:n =
1495 {
1496 \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1497 }
1498 ,pdfcenterwindow / .code:n =
1499 {
1500 \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1501 }
1502 ,pdfcenterwindow / unknown .code:n =
1503 {
1504 \msg_warning:nxxx { hyp } { no-bool }
1505 { pdfcenterwindow }
1506 { \exp_not:n {#1} }
1507 }
1508 ,pdfcenterwindow .default:n = true

```

```

1509 ,pdfdirection .choice:
1510 ,pdfdirection / L2R .code:n =
1511 {
1512     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1513 }
1514 ,pdfdirection / R2L .code:n =
1515 {
1516     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1517 }
1518 ,pdfdirection / .code:n =
1519 {
1520     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1521 }
1522 ,pdfdirection / unknown .code:n =
1523 {
1524     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1525     { pdfdirection }
1526     { L2R , R2L }
1527     { \exp_not:n {#1} }
1528 }
1529 ,pdfdisplaydoctitle .choice:
1530 ,pdfdisplaydoctitle / false .code:n =
1531 {
1532     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1533 }
1534 ,pdfdisplaydoctitle / true .code:n =
1535 {
1536     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1537 }
1538 ,pdfdisplaydoctitle .default:n = true
1539 ,pdfduplex .choices:nn =
1540 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1541 {
1542     \pdf_version_compare:NnTF > {1.6}
1543     {
1544         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1545         { PrintDuplex } { /#1 }
1546     }
1547     {
1548         \msg_warning:nnxx
1549         {hyp}
1550         {ignore-deprecated-or-unknown-option-in-pdf-version}
1551         {pdfduplex}
1552         {\pdf_version:}
1553     }
1554 }%
1555 ,pdfduplex / .code:n =
1556 {
1557     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1558 }
1559 ,pdfduplex / unknown .code:n =
1560 {
1561     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1562     { pdfduplex }

```

```

1563         { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1564         { \exp_not:n {#1} }
1565     }
1566     ,pdffitwindow .choice:
1567     ,pdffitwindow / false .code:n =
1568     {
1569         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1570     }
1571     ,pdffitwindow / true .code:n =
1572     {
1573         \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1574     }
1575     ,pdffitwindow / .code:n =
1576     {
1577         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1578     }
1579     ,pdffitwindow .default:n = true
1580     ,pdffitwindow / unknown .code:n =
1581     {
1582         \msg_warning:nnxx { hyp } { no-bool }
1583         { pdffitwindow }
1584         { \exp_not:n {#1} }
1585     }
1586     ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1587     ,pdflinkmargin .initial:n = {1pt}
1588     ,pdfmenubar .choice:
1589     ,pdfmenubar / true .code:n =
1590     {
1591         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1592     }
1593     ,pdfmenubar / false .code:n =
1594     {
1595         \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1596         { HideMenubar } { true }
1597     }
1598     ,pdfmenubar / .code:n =
1599     {
1600         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1601     }
1602     ,pdfmenubar .default:n = true
1603     ,pdfmenubar / unknown .code:n =
1604     {
1605         \msg_warning:nnxx { hyp } { no-bool }
1606         { pdfmenubar }
1607         { \exp_not:n {#1} }
1608     }
1609     ,pdfnewwindow .choice:
1610     ,pdfnewwindow / true .code:n =
1611     {
1612         \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1613         \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1614     }
1615     ,pdfnewwindow / false .code:n =
1616     {

```

```

1617     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1618     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1619   }
1620 ,pdfnewwindow / .code:n =
1621   {
1622     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1623     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1624   }
1625 ,pdfnonfullscreenpagemode .choices:nn =
1626   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1627   {
1628     \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1629     { NonFullScreenPageMode } {/#1}
1630   }
1631 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1632   {
1633     \pdf_version_compare:NnTF < {1.6}
1634     {
1635       %message
1636     }
1637     {
1638       \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1639       {NonFullScreenPageMode}{/UseAttachments}
1640     }
1641   }
1642 ,pdfnonfullscreenpagemode / .code:n =
1643   {
1644     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1645   }
1646 ,pdfnonfullscreenpagemode / unknown .code:n =
1647   {
1648     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1649     { pdfnonfullscreenpagemode }
1650     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1651     { \exp_not:n {#1} }
1652   }
1653 ,pdfnumcopies .code:n =
1654   {
1655     \pdf_version_compare:NnTF > {1.6}
1656     {
1657       \tl_if_empty:nTF {#1}
1658       {
1659         \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1660       }
1661       {
1662         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1663         {NumCopies}{#1}
1664       }
1665     }
1666     {
1667       \msg_warning:nnxx
1668       {hyp}
1669       {ignore-deprecated-or-unknown-option-in-pdf-version}
1670       {pdfnumcopies}

```

```

1671         {\pdf_version:}
1672     }
1673 }
1674 ,pdfpagelayout .choices:nn =
1675 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1676 { \pdfmanagement_add:nnx {Catalog} { PageLayout }{ /#1 } }
1677 ,pdfpagelayout / .code:n =
1678 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1679 ,pdfpagelayout / unknown .code:n =
1680 {
1681     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1682     { pdfpagelayout }
1683     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1684     { \exp_not:n {#1} }
1685 }
1686 ,pdfpagemode .choices:nn =
1687 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1688 { \pdfmanagement_add:nnx {Catalog} { PageMode }{ /#1 } }
1689 ,pdfpagemode / UseAttachments .code:n =
1690 {
1691     \pdf_version_compare:NnTF > {1.5}
1692     {
1693         \pdfmanagement_add:nnx {Catalog} { PageMode }{ /UseAttachments }
1694     }
1695     {
1696         \msg_warning:nnxx
1697         {hyp}
1698         {ignore-deprecated-or-unknown-value-in-pdf-version}
1699         {UseAttachments}
1700         {\pdf_version:}
1701     }
1702 }
1703 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1704 ,pdfpagemode / unknown .code:n =
1705 {
1706     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1707     { pdfpagemode }
1708     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1709     { \exp_not:n {#1} }
1710 }
1711 ,pdfpagescrop .code:n =
1712 {
1713     \tl_if_empty:nTF {#1} %or blank?
1714     {
1715         \pdfmanagement_remove:nn {Pages} { CropBox }
1716     }
1717     {
1718         \pdfmanagement_add:nnx {Pages} { CropBox } { [#1] }
1719     }
1720 }
1721 ,pdfpicktraybypdfsize .choice:
1722 ,pdfpicktraybypdfsize / true .code:n =
1723 {
1724     \pdf_version_compare:NnTF > {1.6}

```

```

1725     {
1726         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1727         { PickTrayByPDFSize } { true }
1728     }
1729     {
1730         \msg_warning:nnxx
1731         {hyp}
1732         {ignore-deprecated-or-unknown-option-in-pdf-version}
1733         {pdfpicktraybypdfsize}
1734         {\pdf_version:}
1735     }
1736 }
1737 ,pdfpicktraybypdfsize / false .code:n =
1738 {
1739     \pdf_version_compare:NnTF > {1.6}
1740     {
1741         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1742         { PickTrayByPDFSize } { false }
1743     }
1744     {
1745         \msg_warning:nnxx
1746         {hyp}
1747         {ignore-deprecated-or-unknown-option-in-pdf-version}
1748         {pdfpicktraybypdfsize}
1749         {\pdf_version:}
1750     }
1751 }
1752 ,pdfpicktraybypdfsize / .code:n =
1753 {
1754     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1755 }
1756 ,pdfpicktraybypdfsize / unknown .code:n =
1757 {
1758     \msg_warning:nnxx { hyp } { no-bool }
1759     { picktraybypdfsize }
1760     { \exp_not:n {#1} }
1761 }
1762 ,pdfprintarea .choices:nn =
1763 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1764 {
1765     \pdf_version_compare:NnTF < {2.0}
1766     {
1767         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1768         { PrintArea } { /#1 }
1769     }
1770     {
1771         \msg_warning:nnxx
1772         {hyp}
1773         {ignore-deprecated-or-unknown-option-in-pdf-version}
1774         {pdfprintarea}
1775         {\pdf_version:}
1776     }
1777 }%
1778 ,pdfprintarea / .code:n =

```



```

1779     { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1780 ,pdfprintarea / unknown .code:n =
1781 {
1782     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1783     { pdfprintarea }
1784     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1785     { \exp_not:n {#1} }
1786 }
1787 ,pdfprintclip .choices:nn =
1788 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1789 {
1790     \pdf_version_compare:NnTF < {2.0}
1791     {
1792         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1793         { PrintClip } { /#1 }
1794     }
1795     {
1796         \msg_warning:nnxx
1797         {hyp}
1798         {ignore-deprecated-or-unknown-option-in-pdf-version}
1799         {pdfprintclip}
1800         {\pdf_version:}
1801     }
1802 }%
1803 ,pdfprintclip / .code:n =
1804 {
1805     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1806 }
1807 ,pdfprintclip / unknown .code:n =
1808 {
1809     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1810     { pdfprintclip }
1811     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1812     { \exp_not:n {#1} }
1813 }
1814 ,pdfprintpagerange .code:n =
1815 {
1816     \pdf_version_compare:NnTF > {1.6}
1817     {
1818         \tl_if_empty:nTF { #1}
1819         {
1820             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1821             { PrintPageRange }
1822         }
1823         {
1824             \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1825             {PrintPageRange}{[#1]}
1826         }
1827     }
1828     {
1829         \msg_warning:nnxx
1830         {hyp}
1831         {ignore-deprecated-or-unknown-option-in-pdf-version}
1832         {pdfprintpagerange}

```

```

1833         {\pdf_version:}
1834     }
1835 }
1836 ,pdfprintscaling .choices:nn =
1837 { None, AppDefault }
1838 {
1839     \pdf_version_compare:NnTF > {1.5}
1840     {
1841         \pdfmanagement_add:nnx {Catalog / ViewerPreferences }
1842         { PrintScaling } { /#1 }
1843     }
1844     {
1845         \msg_warning:nnxx
1846         {hyp}
1847         {ignore-deprecated-or-unknown-option-in-pdf-version}
1848         {pdfprintscaling}
1849         {\pdf_version:}
1850     }
1851 }%
1852 ,pdfprintscaling / .code:n =
1853 {
1854     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1855 }
1856 ,pdfprintscaling / unknown .code:n =
1857 {
1858     \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1859     { pdfprintarea }
1860     { None, AppDefault }
1861     { \exp_not:n {#1} }
1862 }
1863 ,pdfremotestartview .code:n =
1864 {
1865     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1866     \exp_args:NNV
1867     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1868     {
1869         \tl_set:Nx \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1870     }
1871     {
1872         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1873         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1874     }
1875 }
1876 ,pdfremotestartview .initial:n = {Fit}
1877 % pdfstartpage is special as it shares code with pdfstartview
1878 ,pdfstartpage .code:n =
1879 {
1880     \tl_gset:Nx \g__hyp_dest_pdfstartpage_tl { #1 }
1881     \bool_if:nTF
1882     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1883     {
1884         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1885     }
1886     {

```

```

1887         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1888         {
1889             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1890         }
1891     }
1892 }
1893 ,pdfstartpage .initial:n =1
1894 ,pdfstartview .code:n =
1895 {
1896     \tl_set:Nx \l__hyp_tmpa_tl {#1-null-null-null~}
1897     \exp_args:NNV
1898     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1899     {
1900         \tl_gset:Nx \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1901     }
1902     {
1903         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
1904         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
1905     }
1906     \bool_if:nTF
1907     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1908     {
1909         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1910     }
1911     {
1912         \pdfmanagement_add:nnx {Catalog} { OpenAction }
1913         {
1914             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
1915         }
1916     }
1917 }
1918 ,pdfstartview .initial:n = Fit
1919 ,pdftoolbar .choice:
1920 ,pdftoolbar / true .code:n =
1921 {
1922     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1923 }
1924 ,pdftoolbar / false .code:n =
1925 {
1926     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
1927     { HideToolbar } { true }
1928 }
1929 ,pdftoolbar / true .code:n =
1930 {
1931     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
1932 }
1933 ,pdftoolbar .default:n = true
1934 ,pdftoolbar / unknown .code:n =
1935 {
1936     \msg_warning:nnxx { hyp } { no-bool }
1937     { pdftoolbar }
1938     { \exp_not:n {#1} }
1939 }
1940 % pdfview see below.

```

```

1941 ,pdfviewarea .choices:nn =
1942 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1943 {
1944   \pdf_version_compare:NnTF < {2.0}
1945   {
1946     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1947     { ViewArea } { /#1 }
1948   }
1949   {
1950     \msg_warning:nnxx
1951     {hyp}
1952     {ignore-deprecated-or-unknown-option-in-pdf-version}
1953     {pdfviewarea}
1954     {\pdf_version:}
1955   }
1956 }%
1957 ,pdfviewarea / .code:n =
1958 {
1959   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
1960 }
1961 ,pdfviewarea / unknown .code:n =
1962 {
1963   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1964   { pdfviewarea }
1965   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1966   { \exp_not:n {#1} }
1967 }
1968 ,pdfviewclip .choices:nn =
1969 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1970 {
1971   \pdf_version_compare:NnTF < {2.0}
1972   {
1973     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1974     { ViewClip } { /#1 }
1975   }
1976   {
1977     \msg_warning:nnxx
1978     {hyp}
1979     {ignore-deprecated-or-unknown-option-in-pdf-version}
1980     {pdfviewclip}
1981     {\pdf_version:}
1982   }
1983 }%
1984 ,pdfviewclip / .code:n =
1985 {
1986   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
1987 }
1988 ,pdfviewclip / unknown .code:n =
1989 {
1990   \msg_warning:nnxxx { hyp } { unknown-choice+empty }
1991   { pdfviewclip }
1992   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1993   { \exp_not:n {#1} }
1994 }

```

```

1995 ,pdfwindowui .choice:
1996 ,pdfwindowui / true .code:n =
1997 {
1998   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
1999 }
2000 ,pdfwindowui / false .code:n =
2001 {
2002   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2003   { HideWindowUI } { true }
2004 }
2005 ,pdfwindowui / .code:n =
2006 {
2007   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2008 }
2009 ,pdfwindowui / unknown .code:n =
2010 {
2011   \msg_warning:nnxx { hyp } { no-bool }
2012   { pdfwindowui }
2013   { \exp_not:n {#1} }
2014 }
2015 ,pdfwindowui .default:n = true
2016 }

```

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

```

2017 \keys_define:nn { hyp / setup }
2018 {
2019   ,pdfview .code:n =
2020   {
2021     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2022     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
2023       {
2024         { xyz }
2025         {
2026           \int_compare:nNnTF { \seq_count:N \l__hyp_tmpa_seq } > { 1 }
2027           {
2028             \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2029             \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2030             {
2031               \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2032             }
2033             {
2034               \tl_set:Nx \l__hyp_dest_pdfview_tl
2035               {
2036                 \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2037               }
2038             }
2039           }
2040           {
2041             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2042           }
2043         }
2044         { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2045         { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2046         { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }

```

```

2047 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv}}
2048 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2049 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2050 { fitr }
2051 {
2052   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2053   {
2054     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2055   }
2056   {
2057     %ensure 4 values ...
2058     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2059     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2060     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2061     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2062     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2063     {
2064       \fp_eval:n
2065       {
2066         round
2067         (
2068           abs
2069           (
2070             \seq_item:Nn\l__hyp_tmpa_seq{4}
2071             -
2072             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2073           ),
2074           3
2075         )
2076       }bp
2077     }{}
2078     \box_set_dp:Nn \l__hyp_dest_box
2079     {
2080       \fp_eval:n
2081       {
2082         round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2083       }bp
2084     }
2085     \box_set_ht:Nn \l__hyp_dest_box
2086     {
2087       \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2088     }
2089   }
2090 }
2091 }
2092 {
2093   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2094   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2095 }
2096 }
2097 ,pdfview .initial:n = {xyz}
2098 }

```

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.

```
2099 \keys_define:nn { hyp / setup }
2100 {
2101   ,pdflang .code:n =
2102   {
2103     \tl_if_empty:nTF { #1 }
2104     {
2105       \pdfmanagement_remove:nn {Catalog} { Lang }
2106     }
2107     {
2108       \pdfmanagement_add:nnx {Catalog} { Lang } { (#1) }
2109     }
2110     \__hyp_store_metadata:nn {pdflang}{#1}
2111   }
2112 }
```

12.3.1 “info keys”

`pdfauthor (setup key)` The keys store their value also in the metadata container, so that hyperxmp can use
`pdftitle (setup key)` them. Creator and Producer can’t be removed with the pdfmanagement, but we allow
`pdfcreator (setup key)` to set an empty value.

```
pdfsubject (setup key) 2113 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
pdfproducer (setup key) 2114 {
pdfkeywords (setup key) 2115   \keys_define:nn { hyp / setup }
2116   {
2117     pdf#1 .code:n =
2118     {
2119       \tl_if_blank:nTF {##1}
2120       {
2121         \str_case:nnF { #1 }
2122         {
2123           {creator}
2124           {
2125             \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2126             \pdfmanagement_add:nnx {Info}{Creator}{()}
2127           }
2128           {producer}
2129           {
2130             \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2131             \pdfmanagement_add:nnx {Info}{Producer}{()}
2132           }
2133         }
2134         {
2135           \pdfmanagement_remove:nn {Info}{#2}
2136         }
2137       }
2138     }
2139     \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2140     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
```

```

2141         {
2142             \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2143         }
2144     }
2145     \__hyp_store_metadata:nn {pdf#1}{##1}
2146 }
2147 }
2148 \keys_define:nn { hyp / info }
2149 {
2150     #2 .code:n =
2151     {
2152         \tl_if_blank:nTF {##1}
2153         {
2154             \pdfmanagement_remove:nn {Info}{#2}
2155         }
2156         {
2157             \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2158             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2159             {
2160                 \pdfmanagement_add:nnx {Info}{#2}{\l__hyp_tmpa_str}
2161             }
2162         }
2163         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2164     }
2165     ,unknown .code:n =
2166     {
2167         \__hyp_text_pdfstring_info:nN {##1}\l__hyp_tmpa_str
2168         \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2169         {
2170             \exp_args:Nno
2171             \pdfmanagement_add:nnx {Info}
2172             { \l_keys_key_str } {\l__hyp_tmpa_str}
2173         }
2174     }
2175 }
2176 }
2177 \__hyp_setup_info_key:nn {author} {Author}
2178 \__hyp_setup_info_key:nn {title} {Title}
2179 \__hyp_setup_info_key:nn {producer} {Producer}
2180 \__hyp_setup_info_key:nn {creator} {Creator}
2181 % ignored key: addtopdfcreator
2182 \__hyp_setup_info_key:nn {subject} {Subject}
2183 \__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.

```

2184 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2185 {
2186     \keys_define:nn { hyp / setup }
2187     {
2188         pdf#1 .code:n =
2189         {
2190             \tl_if_blank:nTF {##1}
2191             {

```



```

2192         \pdfmanagement_remove:nn {Info}{#2}
2193     }
2194     {
2195         \pdfmanagement_add:nxx {Info}{#2}{(##1)}
2196     }
2197     \__hyp_store_metadata:nn {pdf#1}{##1}
2198 }
2199 }
2200 \keys_define:nn { hyp / info }
2201 {
2202     #2 .code:n =
2203     {
2204         \tl_if_blank:nTF {##1}
2205         {
2206             \pdfmanagement_remove:nn {Info}{#2}
2207         }
2208         {
2209             \pdfmanagement_add:nxx {Info}{#2}{(##1)}
2210         }
2211         \exp_args:Nx \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2212     }
2213 }
2214 }
2215
2216 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2217 \__hyp_setup_info_date_key:nn {moddate} {ModDate}

```

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

```

2218 \keys_define:nn { hyp / setup }
2219 {
2220     ,pdftrapped .code:n =
2221     {
2222         \exp_args:Nne
2223         \keys_set:nn { hyp / setup } { _pdftrapped = \str_uppercase:n { #1 } }
2224     }
2225     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2226     {
2227         \pdfmanagement_add:nxx {Info}{Trapped}
2228         {/
2229             \str_uppercase:f { \str_head:n { #1 } }
2230             \str_lowercase:f { \str_tail:n { #1 } }
2231         }
2232         \__hyp_store_metadata:nx {pdftrapped}
2233         {
2234             \str_uppercase:f { \str_head:n { #1 } }
2235             \str_lowercase:f { \str_tail:n { #1 } }
2236         }
2237     }
2238     ,_pdftrapped / unknown .code:n =
2239     {
2240         \msg_warning:nxxx { hyp } { unknown-choice }
2241         { pdftrapped }
2242         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2243         { \exp_not:n {#1} }

```

```

2244     }
2245 }

```

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

```

2246 \keys_define:nn { hyp / setup }
2247 {
2248   pdfinfo .code:n =
2249   {
2250     \keys_set:nn { hyp / info } { #1 }
2251   }
2252 }

```

Now we set some default values

```

2253 \keys_set:nn { hyp / setup} {pdfcreator = LaTeX-with-hyperref}
2254 \keys_set:nn { hyp / setup} {pdfauthor  = }
2255 \keys_set:nn { hyp / setup} {pdftitle   = }
2256 \keys_set:nn { hyp / setup} {pdfsubject = }

```

12.4 hyperxmp keys

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

```

2257 \clist_map_inline:nn
2258 {
2259   ,pdfcopyright
2260   ,pdftype
2261   ,pdflicenseurl
2262   ,pdfauthortitle
2263   ,pdfcaptionwriter
2264   ,pdfmetalang
2265   ,pdfapart
2266   ,pdfaconformance
2267   ,pdfuapart
2268   ,pdfxstandard
2269   ,pdfsource
2270   ,pdfdocumentid
2271   ,pdfinstanceid
2272   ,pdfversionid
2273   ,pdfrendition
2274   ,pdfpublication
2275   ,pdfpubtype
2276   ,pdfbytes
2277   ,pdfnumpages
2278   ,pdfissn
2279   ,pdfeissn
2280   ,pdfisbn
2281   ,pdfbookedition
2282   ,pdfpublisher
2283   ,pdfvolumenum
2284   ,pdfissuenum
2285   ,pdfpagerange
2286   ,pdfdoi
2287   ,pdfurl

```

```

2288 ,pdfidentifier
2289 ,pdfsubtitle
2290 ,pdfpubstatus
2291 ,pdfcontactaddress
2292 ,pdfcontactcity
2293 ,pdfcontactregion
2294 ,pdfcontactpostcode
2295 ,pdfcontactcountry
2296 ,pdfcontactphone
2297 ,pdfcontactemail
2298 ,pdfcontacturl
2299 ,pdfdate
2300 }
2301 {
2302   \keys_define:nn { hyp / setup }
2303   {
2304     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2305   }
2306 }
2307
2308 \str_if_eq:eeT {A}{\pdfmeta_standard_item:n{type}}
2309 {
2310   \exp_args:Nnx
2311   \keys_set:nn {hyp / setup}
2312   {
2313     ,pdfapart      = \pdfmeta_standard_item:n {level}
2314     ,pdfaconformance = \pdfmeta_standard_item:n {conformance}
2315   }
2316 }

```

12.5 Transitions

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

```

2317 \keys_define:nn { hyp / setup }
2318 {
2319   pdfpageduration .code:n =
2320   {
2321     \tl_if_blank:nTF { #1 }
2322     {
2323       \pdfmanagement_remove:nn {Page}{Dur}
2324     }
2325     {
2326       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2327     }
2328   }
2329 }

```

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

<code>style</code>	one of Split, Blinds, Box, Wipe, Dissolve, Glitter, R, Fly, Push, Cover, Uncover, Fade
<code>duration</code>	a number, describes the duration of the transition
<code>direction</code>	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)
<code>motion</code>	one of I, O, only relevant for Split, Box and Fly
<code>scale</code>	a number, only relevant for Fly style
<code>opaque</code>	true or false, only relevant for Fly style

```
2330 \keys_define:nn { hyp / setup }
2331 {
2332   pdfpagetransition .code:n =
2333   {
2334     \tl_if_blank:nTF {#1}
2335     {
2336       \pdfmanagement_remove:nn {Page}{Trans}
2337     }
2338     {
2339       \group_begin:
2340       \keys_set:nn { hyp / trans }{style=R,#1}
2341       \pdf_object_unnamed_write:nx { dict }
2342       {
2343         \pdfdict_use:n {l__hyp_page/Trans}
2344       }
2345       \pdfmanagement_add:nnx {Page}{Trans}{\pdf_object_ref_last:}
2346       \group_end:
2347     }
2348   }
2349 }
2350 \keys_define:nn { hyp / trans }
2351 {
2352   ,style .choices:nn =
2353   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2354   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2355   ,style / unknown .code:n =
2356   {
2357     \msg_warning:nnxxx { hyp } { unknown-choice }
2358     { trans / style }
2359     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2360     { \exp_not:n {#1} }
2361   }
```

```

2362 ,duration .code:n =
2363 {
2364     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{#1}
2365 }
2366 ,direction .choices:nn =
2367 {H,V}
2368 { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2369 ,direction .choices:nn =
2370 {0,90,180,270,315}
2371 { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2372 ,direction / None .code:n =
2373 { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2374 ,direction / unknown .code:n =
2375 {
2376     \msg_warning:nnxxx { hyp } { unknown-choice }
2377     { trans / direction }
2378     {
2379         H~(horizontal,~only~Split,~Blinds),
2380         V~(vertical,~only~Split,~Blinds),
2381         0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2382         90~(bottom~to~top,~only~Wipe),
2383         180~(right~to~left,~only~Wipe),
2384         270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2385         315~(top~left~to~bottom,~only~Glitter),
2386         None~(only~Fly)
2387     }
2388     { \exp_not:n {#1} }
2389 }
2390 ,motion .choices:nn =
2391 {I,0}
2392 { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2393 ,motion / unknown .code:n =
2394 {
2395     \msg_warning:nnxxx { hyp } { unknown-choice }
2396     { trans / motion }
2397     { I~(inwards) , 0~(outwards) }
2398     { \exp_not:n {#1} }
2399 }
2400 ,scale .code:n =
2401 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2402 ,opaque .choices:nn = {true,false}
2403 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2404 ,opaque / unknown .code:n =
2405 {
2406     \msg_warning:nnxxx { hyp } { unknown-choice }
2407     { trans / B }
2408     { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2409     { \exp_not:n {#1} }
2410 }
2411 % try to set unknown keys as style
2412 ,unknown .code:n =
2413 {
2414     % warning ...
2415     \exp_args:Nnx\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }

```

```

2416     }
2417 }
    Finally we process the package option list, to get most keys working
2418 \keys_set_known:nv{hyp/setup}{opt@hyperref.sty}
    Unfinished Form field code
2419 \NewDocumentCommand \MakeFieldObject { m m }
2420 {
2421     \pdfxform_new:nnn { #2 }{} { #1 }
2422 }
2423
2424
2425 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2426 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2427
2428 \let\HyField@afields\ltx@empty
2429 \let\HyField@cofields\ltx@empty
2430 %% UF test for old pdftex removed
2431 %\let\HyField@AuxAddToFields\ltx@gobble
2432 %\let\HyField@AuxAddToCoFields\ltx@gobbletwo
2433 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2434
2435 % the value doesn't matter, but with a prop we avoid duplicates and it is
2436 % clearly faster than removing them from a sequence
2437 \def\HyField@AuxAddToFields#1
2438 {
2439     \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2440 }%
2441
2442 %fields with empty key get a value too -- lets hope that
2443 %this give the expected behaviour
2444 \def\HyField@AuxAddToCoFields #1 #2
2445 {
2446     \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2447 }
2448
2449 \Hy@AtBeginDocument
2450 {
2451     \if@files
2452         \immediate\write\@mainaux{%
2453             \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2454         }%
2455         \immediate\write\@mainaux{%
2456             \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2457         }%
2458     \fi
2459     \let\HyField@AfterAuxOpen\@firstofone
2460 }%
2461
2462 \def\HyField@AddToFields
2463 {
2464     \exp_args:Nx\HyField__hypAddToFields
2465     {
2466         \pdfannot_box_ref_last:

```

```

2467     }
2468     \ifx\Fld@calculate@code\ltx@empty
2469     \else
2470     \begingroup
2471     \Hy@safe@activetrue
2472     \edef\Hy@temp{%
2473     \endgroup
2474     \if@filesw
2475     \write\@mainaux
2476     {
2477     \string\HyField@AuxAddToCoFields
2478     {
2479     \Fld@calculate@sortkey
2480     }
2481     {
2482     \pdfannot_box_ref_last:
2483     }
2484     }
2485     \fi
2486     }%
2487     \Hy@temp
2488     \fi
2489     }%
2490
2491     \def\HyField__hypAddToFields#1{
2492     \HyField@AfterAuxOpen{%
2493     \if@filesw
2494     \write\@mainaux{%
2495     \string\HyField@AuxAddToFields{#1}%
2496     }%
2497     \fi
2498     }%
2499     }%
2500
2501     \ExplSyntaxOff
2502     \ExplSyntaxOn
2503
2504     \def\@Form[#1]
2505     {
2506     \kvsetkeys{Form}{#1}
2507     \pdf@ifdraftmode{}
2508     {
2509     \Hy@FormObjects
2510     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2511     {
2512     \pdfmanagement_add:nx { Catalog / AcroForm } { Fields }{##1}
2513     %\pdfmanagement_show:n { Catalog / AcroForm }
2514     }
2515     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2516     {
2517     \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2518     {
2519     \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2520     }

```

```

2521         \seq_sort:Nn \l__hyp_tmpa_seq
2522         {
2523             \str_compare:nNnTF {##1} > {##2}
2524             { \sort_return_swapped: }
2525             { \sort_return_same: }
2526         }
2527     \seq_map_inline:Nn \l__hyp_tmpa_seq
2528     {
2529         \pdfmanagement_add:nnx { Catalog / AcroForm }
2530         { CO }
2531         {
2532             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2533         }
2534     }
2535 }
2536 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2537 {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2538 \pdfmanagement_add:nnx {Catalog / AcroForm/DR/Font }
2539 {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2540 \pdfmanagement_add:nnx {Catalog /AcroForm}
2541 {DA}{(/Helv~10~Tf~0~g)}
2542 \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2543 {
2544     \legacy_if:nT { HyField@NeedAppearances }
2545     {
2546         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2547     }
2548 }
2549 {
2550     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2551 }
2552 }
2553 }
2554 \ExplSyntaxOff
2555 \let\@endForm\ltx@empty
2556 \let\HyAnn@AbsPageLabel\ltx@empty
2557 \let\Fld@pageobjref\ltx@empty
2558
2559 \ExplSyntaxOn
2560 \newcount\HyAnn@Count
2561 \HyAnn@Count=\ltx@zero
2562 \def\HyAnn@AbsPageLabel
2563 {
2564     \global\advance\HyAnn@Count by\ltx@one
2565     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2566     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2567     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2568     \__hyp_ref_label:en {HyAnn@\the\HyAnn@Count}{abspage}
2569     \__hyp_ref_check:en {HyAnn@\the\HyAnn@Count}{abspage}
2570 }%
2571 \def\Fld@pageobjref
2572 {
2573     \__hyp_ref_if_exist:enT {HyAnn@\the\HyAnn@Count}{abspage}
2574     {

```



```

2575         /P~\pdf_pageobject_ref:n
2576         {
2577             \__hyp_ref_value:en{HyAnn@\the\HyAnn@Count}{abspage}
2578         }
2579     }
2580 }
2581 \ExplSyntaxOff
2582 \ExplSyntaxOn
2583 %% check if the attr should be set through
2584 %% hooks.
2585 %% check if options are missing.
2586 \def\@TextField[#1]#2{% parameters, label
2587     \def\Fld@name{#2}%
2588     \let\Fld@default\ltx@empty
2589     \let\Fld@value\@empty
2590     \def\Fld@width{\DefaultWidthofText}%
2591     \def\Fld@height{%
2592         \ifFld@multiline
2593             \DefaultHeightofTextMultiline
2594         \else
2595             \DefaultHeightofText
2596         \fi
2597     }%
2598     \begingroup
2599         \expandafter\HyField@SetKeys\expandafter{%
2600             \DefaultOptionsofText,#1%
2601         }%
2602         \PDFForm@Name
2603         \HyField@FlagsText
2604         \ifFld@hidden\def\Fld@width{1sp}\fi
2605         \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2606         \LayoutTextField{#2}{%
2607             \leavevmode
2608             \HyAnn@AbsPageLabel
2609             \Hy@escapeform\PDFForm@Text
2610             \pdfannot_box:nnnn
2611                 {\Fld@width}
2612                 {\Fld@height}
2613                 {0pt} %is this correct?
2614                 {\PDFForm@Text}
2615             \MakeTextField{\Fld@width}{\Fld@height}
2616             \HyField@AddToFields
2617         }%
2618     \endgroup
2619 }
2620 \providecommand\@curropt{}
2621 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2622     \def\Fld@name{#2}
2623     \let\Fld@default\relax
2624     \let\Fld@value\relax
2625     \def\Fld@width{\DefaultWidthofChoiceMenu}
2626     \def\Fld@height{\DefaultHeightofChoiceMenu}
2627     \begingroup
2628         \Fld@menulength=0 %

```

```

2629 \@tempdima\z@
2630 \clist_map_variable:nNn { #3 } \@curropt
2631 %\@for\@curropt:=#3\do
2632 {%
2633   \expandafter\Fld@checkequals\@curropt==\\%
2634   \Hy@StepCount\Fld@menulength
2635   \settowidth{\@tempdimb}{\@currDisplay}%
2636   \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2637 }%
2638 \advance\@tempdima by~15\p@
2639 \beginngroup
2640 \HyField@SetKeys{#1}
2641 \edef\x{\endgroup
2642   \noexpand\expandafter
2643   \noexpand\HyField@SetKeys
2644   \noexpand\expandafter{%
2645     \expandafter\noexpand\csname DefaultOptionsof%
2646       \ifFld@radio
2647         Radio%
2648       \else
2649         \ifFld@combo
2650           \ifFld@popdown
2651             PopdownBox%
2652           \else
2653             ComboBox%
2654           \fi
2655         \else
2656           ListBox%
2657         \fi
2658       \fi
2659     \endcsname
2660   }%
2661 }\x
2662 \HyField@SetKeys{#1}%
2663 \PDFForm@Name
2664 \ifFld@hidden\def\Fld@width{1sp}\fi
2665 \ifx\Fld@value\relax
2666   \let\Fld@value\Fld@default
2667 \fi
2668 \LayoutChoiceField{#2}{%
2669   \ifFld@radio
2670     \HyField@FlagsRadioButton
2671     \__hypRadio{#3}%
2672   \else
2673     \beginngroup
2674     \HyField@FlagsChoice
2675     \ifdim\Fld@width<\@tempdima
2676       \ifdim\@tempdima<1cm\@tempdima1cm\fi
2677       \edef\Fld@width{\the\@tempdima}%
2678     \fi
2679     \ifFld@combo
2680     \else
2681       \@tempdima=\the\Fld@menulength\Fld@charsize
2682       \advance\@tempdima by~\Fld@borderwidth bp %

```

```

2683         \advance\@tempdima by~\Fld@borderwidth bp %
2684         \edef\Fld@height{\the\@tempdima}%
2685         \fi
2686         \__hypListbox{#3}%
2687     \endgroup
2688 \fi
2689 }%
2690 \endgroup
2691 }
2692
2693 \def\__hypRadio#1{%
2694     \Fld@listcount=0~%
2695     %\show\Fld@default
2696     \EdefEscapeName\Fld@default{\Fld@default}%
2697     \clist_map_variable:nNn { #1 } \@curropt
2698     %\@for\@curropt:=#1\do
2699     {%
2700         \expandafter\Fld@checkequals\@curropt==\%
2701         \EdefEscapeName\@currValue{\@currValue}%
2702         \Hy@StepCount\Fld@listcount
2703         \@currDisplay\space
2704         \leavevmode
2705         \HyAnn@AbsPageLabel
2706         \Hy@escapeform\PDFForm@Radio
2707         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2708         {
2709             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2710             {
2711                 \group_begin:
2712                 \fontfamily{pzd}
2713                 \fontencoding{U}
2714                 \fontseries{m}
2715                 \fontshape{n}
2716                 \selectfont
2717                 \char123
2718                 \group_end:
2719             }
2720         }
2721         \pdfannot_box:nnnx
2722         {\Fld@width}
2723         {\Fld@height}
2724         {0pt} %is this correct?
2725         {
2726             \PDFForm@Radio
2727             /AP
2728             <<
2729             /N
2730             <<
2731             /@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2732             %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2733             >>
2734             >>
2735         }
2736         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }

```

```

2737     \int_compare:nNtT { \Fld@listcount } = { 1 }
2738     { \HyField@AddToFields }
2739     \c_space_tl % deliberate space between radio buttons
2740                 % to do: --> should be configurable
2741 }%
2742 }
2743
2744 \newcount\Fld@listcount
2745 \def\__hypListbox#1
2746 {
2747     \HyField@PDFChoices{#1}
2748     \mode_leave_vertical:
2749     \HyAnn@AbsPageLabel
2750     \Hy@escapeform\PDFForm@List
2751     \pdf_link_user:nnn
2752         {widget} %perhaps we need more types??
2753         {\PDFForm@List}
2754         {\MakeChoiceField{\Fld@width}{\Fld@height}}
2755     \HyField@AddToFields
2756 }
2757
2758
2759 \def\@PushButton[#1]#2{% parameters, label
2760     \def\Fld@name{#2}%
2761     \group_begin:
2762         \exp_args:No\HyField@SetKeys
2763         {
2764             \DefaultOptionsofPushButton,#1
2765         }
2766         \PDFForm@Name
2767         \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2768         {
2769             \HyField@FlagsPushButton
2770             \legacy_if:nT {Fld@hidden}
2771             {
2772                 \def\Fld@width{1sp}
2773             }
2774             \LayoutPushButtonField
2775             {
2776                 \mode_leave_vertical:
2777                 \HyAnn@AbsPageLabel
2778                 \Hy@escapeform\PDFForm@Push
2779                 \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2780                 \pdfannot_box:nnnn
2781                 {\box_wd:N\l_tmpa_box}
2782                 {\box_ht:N\l_tmpa_box}
2783                 {\box_dp:N\l_tmpa_box} %is this correct?
2784                 {\PDFForm@Push}
2785                 {\box_use:N\l_tmpa_box}
2786                 \HyField@AddToFields
2787             }
2788         }
2789     {
2790         \msg_error:nn { hyp }{ pdfa-no-push-button }

```

```

2791         \LayoutPushButtonField
2792         {
2793             \mode_leave_vertical:
2794             \MakeButtonField{#2}
2795         }
2796     }
2797 \group_end:
2798 }
2799
2800 \def\@Submit[#1]#2
2801 {
2802     \def\Fld@width {\DefaultWidthofSubmit}
2803     \def\Fld@height{\DefaultHeightofSubmit}
2804     \group_begin:
2805         \exp_args:No\HyField@SetKeys
2806         {
2807             \DefaultOptionsofSubmit,#1
2808         }
2809     \HyField@FlagsPushButton
2810     \HyField@FlagsSubmit
2811     \legacy_if:nT { Fld@hidden }
2812     {
2813         \def\Fld@width{1sp}
2814     }
2815     \mode_leave_vertical:
2816     \HyAnn@AbsPageLabel
2817     \Hy@escapeform\PDFForm@Submit
2818     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2819     \pdfxform_if_exist:nF
2820     { __hyp_xform_Submit }
2821     {
2822         \pdfxform_new:nnn { __hyp_xform_Submit }{}
2823         {
2824             \fbox{\color_select:n{yellow}\textsf{Submit}}
2825         }
2826         \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2827         {
2828             \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2829         }
2830     }
2831     \pdfannot_box:nnnn
2832     {\box_wd:N\l_tmpa_box}
2833     {\box_ht:N\l_tmpa_box}
2834     {\box_dp:N\l_tmpa_box} %is this correct?
2835     {
2836         \PDFForm@Submit
2837         /AP<<
2838         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2839         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2840         >>
2841     }
2842     \HyField@AddToFields
2843     \box_use:N\l_tmpa_box
2844

```

```

2845     \group_end:
2846 }
2847
2848 \def\@Reset[#1]#2
2849 {
2850     \def\Fld@width {\DefaultWidthofReset}
2851     \def\Fld@height{\DefaultHeightofReset}
2852     \group_begin:
2853     \exp_args:No\HyField@SetKeys
2854     {
2855         \DefaultOptionsofReset,#1
2856     }
2857     \mode_leave_vertical:
2858     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2859     {
2860         \HyField@FlagsPushButton
2861         \legacy_if:nT { Fld@hidden }
2862         { \def\Fld@width{1sp} }
2863         \HyAnn@AbsPageLabel
2864         \Hy@escapeform\PDFForm@Reset
2865         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2866         \pdfannot_box:nnnn
2867         {\box_wd:N\l_tmpa_box}
2868         {\box_ht:N\l_tmpa_box}
2869         {\box_dp:N\l_tmpa_box} %is this correct?
2870         { \PDFForm@Reset }
2871         \HyField@AddToFields
2872         \box_use:N \l_tmpa_box
2873     }
2874     {
2875         \msg_error:nn { hyp }{ pdfa-no-reset-button }
2876         \MakeButtonField{#2}
2877     }
2878     \group_end:
2879 }
2880
2881 \def\@CheckBox[#1]#2
2882 {% parameters, label
2883     \def\Fld@name{#2}
2884     \def\Fld@default{0}
2885     \group_begin:
2886     \def\Fld@width {\DefaultWidthofCheckBox}
2887     \def\Fld@height{\DefaultHeightofCheckBox}
2888     \exp_args:No\HyField@SetKeys
2889     {
2890         \DefaultOptionsofCheckBox,#1
2891     }
2892     \PDFForm@Name
2893     \HyField@FlagsCheckBox
2894     \legacy_if:nT { Fld@hidden }
2895     {
2896         \def\Fld@width{1sp}
2897     }
2898     \LayoutCheckField{#2}

```

```

2899 {
2900   \mode_leave_vertical:
2901   \HyAnn@AbsPageLabel
2902   \Hy@escapeform\PDFForm@Check
2903   \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
2904   {
2905     \pdfxform_new:nnn
2906     {__hyp_xform_CheckMarkYes}{-}
2907     {
2908       \group_begin:
2909       \fontfamily{pzd}
2910       \fontencoding{U}
2911       \fontseries{m}
2912       \fontshape{n}
2913       \selectfont
2914       \char51
2915       \group_end:
2916     }
2917     \pdfxform_new:nnn
2918     {__hyp_xform_CheckMarkOff}{-}
2919     {
2920       \group_begin:
2921       \fontfamily{pzd}
2922       \fontencoding{U}
2923       \fontseries{m}
2924       \fontshape{n}
2925       \selectfont
2926       \phantom{\char51} %perhaps xetex needs some small glyph ..
2927       \group_end:
2928     }
2929   }
2930   \pdfannot_box:nnnn
2931   {\Fld@width}
2932   {\Fld@height}
2933   {Opt} %is this correct?
2934   {\PDFForm@Check}
2935   \HyField@AddToFields %check if this works with xelatex ...
2936 }
2937 \group_end:
2938 }
2939 \ExplSyntaxOff
2940
2941 %hm. Should a luatex driver use type1 fonts in fields????
2942 \ExplSyntaxOn
2943 \def\Hy@FormObjects
2944 {
2945   \pdf_object_new:nn {__hyp/Encoding/pdfdoc } { dict }
2946   \pdf_object_new:nn {__hyp/Font/ZaDb } { dict }
2947   \pdf_object_new:nn {__hyp/Font/Helv } { dict }
2948   \pdf_object_write:nx {__hyp/Encoding/pdfdoc }
2949   {
2950     /Type/Encoding
2951     /Differences[
2952       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek

```

```

2953     /ring/tilde
2954     \c_space_tl
2955     39/quotesingle
2956     \c_space_tl
2957     96/grave %
2958     \iow_newline:
2959     128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
2960     /fraction/guilsinglleft/guilsinglright/minus/perthousand
2961     /quotedblbase/quotedblleft/quotedblright/quoteleft
2962     /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
2963     /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
2964     \iow_newline:
2965     164/currency
2966     \c_space_tl
2967     166/brokenbar
2968     \c_space_tl
2969     168/dieresis/copyright/ordfeminine
2970     \c_space_tl
2971     172/logicalnot/.notdef/registered/macron/degree/plusminus
2972     /twosuperior/threesuperior/acute/mu
2973     \c_space_tl
2974     183/periodcentered/cedilla/onesuperior/ordmasculine
2975     \c_space_tl
2976     188/onequarter/onehalf/threequarters
2977     \iow_newline:
2978     192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
2979     /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
2980     /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
2981     /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
2982     /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
2983     /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
2984     /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
2985     /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
2986     /ocircumflex/otilde/odieresis/divide/oshlash/ugrave
2987     /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
2988   ]
2989 }
2990 \pdf_object_write:nn {__hyp/Font/ZaDb }
2991 {
2992   /Type/Font
2993   /Subtype/Type1
2994   /Name/ZaDb
2995   /BaseFont/ZapfDingbats
2996 }
2997 \pdf_object_write:nx {__hyp/Font/Helv }
2998 {
2999   /Type/Font
3000   /Subtype/Type1
3001   /Name/Helv
3002   /BaseFont/Helvetica
3003   /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3004 }
3005 \global\let\Hy@FormObjects\relax
3006 }

```



```

3007 \ExplSyntaxOff
3008 \providecommand*{\Fld@pageobjref}{\}
3009 \ifcsname pdf@escapestring\endcsname
3010   \def\Hy@escapeform#1{%
3011     \ifHy@pdfescapeform
3012       \let\Hy@escapestring\pdfescapestring
3013     \else
3014       \let\Hy@escapestring\@firstofone
3015     \fi
3016   }%
3017   \Hy@escapeform{\}%
3018 \else
3019   \let\Hy@escapestring\@firstofone
3020   \def\Hy@escapeform#1{%
3021     \ifHy@pdfescapeform
3022       \def\Hy@escapestring##1{%
3023         \noexpand\Hy@escapestring{\noexpand##1}%
3024       }%
3025       \edef\Hy@temp{#1}%
3026       \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{\}\@nil
3027       \def\Hy@escapestring##1{%
3028         \ifundefined{Hy@esc@\string##1}{%
3029           ##1%
3030           \ThisShouldNotHappen
3031         }{%
3032           \csname Hy@esc@\string##1\endcsname
3033         }%
3034       }%
3035     \else
3036       \let\Hy@escapestring\@firstofone
3037     \fi
3038   }%
3039   \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3040     \ifx\#3\%
3041     \else
3042       \expandafter
3043       \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3044       \ltx@ReturnAfterFi{%
3045         \Hy__hypescapeform#3\@nil
3046       }%
3047     \fi
3048   }%
3049 \fi
3050 \def\PDFForm@Name{%
3051   \PDFForm__hypName\Fld@name
3052   \ifx\Fld@altname\relax
3053   \else
3054     \PDFForm__hypName\Fld@altname
3055   \fi
3056   \ifx\Fld@mappingname\relax
3057   \else
3058     \PDFForm__hypName\Fld@mappingname
3059   \fi
3060 }

```

```

3061 \def\PDFForm__hypName#1{%
3062   \begingroup
3063     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3064       \ifHy@unicode
3065         \Hy@unicodedefalse
3066       \fi
3067     \fi
3068     \pdfstringdef\Hy@gtemp#1%
3069   \endgroup
3070   \let#1\Hy@gtemp
3071 }
3072 \def\Fld@X@additionalactions{%
3073   \ifx\Fld@keystroke@code\@empty
3074   \else
3075     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3076   \fi
3077   \ifx\Fld@format@code\@empty
3078   \else
3079     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3080   \fi
3081   \ifx\Fld@validate@code\@empty
3082   \else
3083     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3084   \fi
3085   \ifx\Fld@calculate@code\@empty
3086   \else
3087     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3088   \fi
3089   \ifx\Fld@onfocus@code\@empty
3090   \else
3091     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3092   \fi
3093   \ifx\Fld@onblur@code\@empty
3094   \else
3095     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3096   \fi
3097   \ifx\Fld@onmousedown@code\@empty
3098   \else
3099     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3100   \fi
3101   \ifx\Fld@onmouseup@code\@empty
3102   \else
3103     /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3104   \fi
3105   \ifx\Fld@onenter@code\@empty
3106   \else
3107     /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3108   \fi
3109   \ifx\Fld@onexit@code\@empty
3110   \else
3111     /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3112   \fi
3113 }
3114 \ExplSyntaxOn

```

```

3115 \def\Fld@additionalactions
3116 {%
3117   \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{}
3118   {
3119     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3120     {/AA<<\Fld@X@additionalactions>>}
3121   }
3122 }
3123 \ExplSyntaxOff
3124 \def\Fld@annotnames{%
3125   /T(\Fld@name)%
3126   \ifx\Fld@altname\relax
3127   \else
3128     /TU(\Fld@altname)%
3129   \fi
3130   \ifx\Fld@mappingname\relax
3131   \else
3132     /TM(\Fld@mappingname)%
3133   \fi
3134 }
3135 \ExplSyntaxOn
3136 \def\PDFForm@Check
3137 {
3138   /Subtype/Widget
3139   ~\Fld@annotflags
3140   ~\Fld@pageobjref
3141   ~\Fld@annotnames
3142   /FT/Btn
3143   \Fld@flags
3144   /Q~\Fld@align
3145   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3146   /AP
3147   <<
3148   /N
3149   <<
3150     /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3151     /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3152   >>
3153   >>
3154   /MK<<
3155     \int_compare:nNnF {\Fld@rotation}={0}
3156     {
3157       /R~\Fld@rotation
3158     }
3159     \tl_if_empty:NF\Fld@bordercolor
3160     {
3161       /BC[\Fld@bordercolor]
3162     }
3163     \tl_if_empty:NF\Fld@bcolor
3164     {
3165       /BG[\Fld@bcolor]
3166     }
3167     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3168   >>

```

```

3169 /DA
3170 (
3171   /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3172   \tl_if_empty:NF \Fld@color
3173   {
3174     \c_space_tl \Fld@color
3175   }
3176 )
3177 /H/P
3178 \legacy_if:nTF {Fld@checked}
3179 {
3180   /V/Yes /AS/Yes
3181 }
3182 {
3183   /V/Off /AS/Off
3184 }
3185 \Fld@additionalactions
3186 }
3187 \ExplSyntaxOff
3188 \ExplSyntaxOn
3189 \def\PDFForm@Push
3190 {
3191   /Subtype/Widget
3192   ~\Fld@annotflags
3193   ~\Fld@pageobjref
3194   ~\Fld@annotnames
3195   /FT/Btn
3196   ~\Fld@flags
3197   /H/P
3198   /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3199   \bool_if:nT
3200   {
3201     !\int_compare_p:nNn {\Fld@rotation} = {0}
3202     ||
3203     \tl_if_exist_p:N \Fld@bordercolor
3204   }
3205   {
3206     /MK
3207     <<
3208       \int_compare:nNf {\Fld@rotation} = {0}
3209       {
3210         /R~\Fld@rotation
3211       }
3212       \tl_if_exist:NT \Fld@bordercolor
3213       {
3214         /BC[\Fld@bordercolor]
3215       }
3216     >>
3217   }
3218   /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3219   \Fld@additionalactions
3220 }
3221
3222 \ExplSyntaxOff

```

```

3223 \def\PDFForm@List{%
3224 /Subtype/Widget%
3225 \Fld@annotflags
3226 \Fld@pageobjref
3227 \Fld@annotnames
3228 /FT/Ch%
3229 \Fld@flags
3230 /Q \Fld@align
3231 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3232 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3233     \ifx\Fld@bordercolor\relax\else 1\fi
3234     \ifx\Fld@bcolor\relax \else 1\fi
3235     \space
3236 \else
3237     /MK<<%
3238     \ifnum\Fld@rotation=\z@
3239     \else
3240     /R \Fld@rotation
3241     \fi
3242     \ifx\Fld@bordercolor\relax
3243     \else
3244     /BC[\Fld@bordercolor]%
3245     \fi
3246     \ifx\Fld@bcolor\relax
3247     \else
3248     /BG[\Fld@bcolor]%
3249     \fi
3250     >>%
3251 \fi
3252 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3253     \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3254 \Fld@choices
3255 \Fld@additionalactions
3256 }
3257 \ExplSyntaxOn
3258 \def\PDFForm@Radio
3259 {
3260 /Subtype/Widget
3261 ~\Fld@annotflags
3262 ~\Fld@pageobjref
3263 ~\Fld@annotnames
3264 /FT/Btn
3265 \Fld@flags
3266 /H/P
3267 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3268 /MK<<
3269     \ifnum\Fld@rotation=\z@
3270     \else
3271     /R~\Fld@rotation
3272     \fi
3273     \ifx\Fld@bordercolor\relax
3274     \else
3275     /BC[\Fld@bordercolor]%
3276     \fi

```

```

3277 \ifx\Fld@bcolor\relax
3278 \else
3279 /BG[\Fld@bcolor]%
3280 \fi
3281 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3282 >>
3283 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3284 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3285 \ifx\Fld@default\@empty
3286 /V/Off%
3287 /DV/Off%
3288 \else
3289 /V/\Fld@default
3290 /DV/\Fld@default
3291 \fi
3292 \Fld@additionalactions
3293 }
3294 \ExplSyntaxOff
3295 \ExplSyntaxOn
3296 % Does an appeareance dict make sense here?
3297 \def\PDFForm@Text
3298 {
3299 /Subtype/Widget
3300 ~\Fld@annotflags
3301 ~\Fld@pageobjref
3302 ~\Fld@annotnames
3303 /FT/Tx
3304 ~\Fld@flags
3305 /Q~\Fld@align
3306 /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3307 \bool_if:nT
3308 {
3309 !\int_compare_p:nNn {\Fld@rotation} = {0}
3310 ||
3311 \tl_if_exist_p:N \Fld@bordercolor
3312 ||
3313 \tl_if_exist_p:N \Fld@bcolor
3314 }
3315 {
3316 /MK
3317 <<
3318 \int_compare:nNnF {\Fld@rotation} = {0}
3319 {
3320 /R~\Fld@rotation
3321 }
3322 \tl_if_exist:NT \Fld@bordercolor
3323 {
3324 /BC[\Fld@bordercolor]
3325 }
3326 \tl_if_exist:NT \Fld@bcolor
3327 {
3328 /BG[\Fld@bcolor]
3329 }
3330 >>

```

```

3331     }
3332 /DA
3333 (
3334     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3335     \tl_if_empty:NF {\c_space_tl\Fld@color}
3336 )
3337 /DV(\Hy@escapestring{\Fld@default})
3338 /V(\Hy@escapestring{\Fld@value})
3339 ~\Fld@additionalactions
3340 \int_compare:nNnT { \Fld@maxlen}>{0}
3341 {
3342     /MaxLen~\Fld@maxlen
3343 }
3344 }
3345 \ExplSyntaxOff
3346
3347 \def\PDFForm@Submit{%
3348     /Subtype/Widget%
3349     \Fld@annotflags
3350     \Fld@pageobjref
3351     \Fld@annotnames
3352     /FT/Btn%
3353     \Fld@flags
3354     /H/P%
3355     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3356     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3357         \ifx\Fld@bordercolor\relax\else 1\fi
3358         \space
3359 \else
3360     /MK<<%
3361         \ifnum\Fld@rotation=\z@
3362         \else
3363             /R \Fld@rotation
3364         \fi
3365         \ifx\Fld@bordercolor\relax
3366         \else
3367             /BC[\Fld@bordercolor]%
3368         \fi
3369     >>%
3370 \fi
3371 /A<<%
3372     /S/SubmitForm%
3373     /F<<%
3374         /FS/URL%
3375         /F(\Hy@escapestring{\Form@action})%
3376     >>%
3377     \Fld@submitflags
3378 >>%
3379 \Fld@additionalactions
3380 }
3381 \ExplSyntaxOn
3382 \def\PDFForm@Reset{%
3383     /Subtype/Widget%
3384     \Fld@annotflags

```

```

3385 \Fld@pageobjref
3386 \Fld@annotnames
3387 /FT/Btn%
3388 \Fld@flags
3389 /H/P%
3390 /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3391 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3392 \ifx\Fld@bordercolor\relax\else 1\fi
3393 \space
3394 \else
3395 /MK<<%
3396 \ifnum\Fld@rotation=\z@
3397 \else
3398 /R~\Fld@rotation
3399 \fi
3400 \ifx\Fld@bordercolor\relax
3401 \else
3402 /BC[\Fld@bordercolor]%
3403 \fi
3404 >>%
3405 \fi
3406 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3407 /A<</S/ResetForm>>%
3408 \Fld@additionalactions
3409 }%
3410
3411
3412 %these patterns are used in hyperref checks.
3413 %it is unclear if they are really useful and if a backend support is
3414 %needed.
3415 \str_case:VnF \c_sys_backend_str
3416 {
3417 { pdfmode }
3418 {
3419 \def\HyPat@ObjRef
3420 {
3421 [0-9]*[1-9][0-9]*~0~R
3422 }
3423 }
3424 { dvipdfmx }
3425 {
3426 \def\HyPat@ObjRef
3427 {
3428 @[\~]+
3429 }
3430 }
3431 { xdvipdfmx }
3432 {
3433 \def\HyPat@ObjRef
3434 {
3435 @[\~]+
3436 }
3437 }
3438 }

```



```

3439 { %also set in hyperref sty, so probably not needed.
3440 \def\HyPat@ObjRef/{.+}
3441 }
3442
3443
3444 \ExplSyntaxOff
3445 % UF: removed Hy@writebookmark
3446 % \Hy@currentbookmarklevel{0}
3447 % \Hy@numberline
3448 % \__hypwritetorep
3449 % counter{bookmark@seq@number}
3450 % removed \HyPsd@SanitizeForOutFile, not needed
3451 % removed \currentpdfbookmark, defined by bookmark,
3452 % should use \newcommand there
3453 % removed \subpdfbookmark, defined by bookmark,
3454 % should use \newcommand there
3455 % removed \belowpdfbookmark, defined by bookmark,
3456 % should use \newcommand there
3457 % removed \pdfbookmark, defined by bookmark,
3458 % \BOOKMARK
3459 % \@BOOKMARK
3460 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3461 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3462 %% removed \ReadBookmarks / unneeded with bookmark.
3463 %% removed \Hy@OutlineName
3464 %% removed \check@bm@number
3465 %% removed \calc@bm@number
3466
3467 \ifHy@implicit
3468 \else
3469 \expandafter\endinput
3470 \fi
3471 \newlength\Hy@SectionHShift
3472 \def\Hy@SectionAnchorHref#1{%
3473 \ifx\protect\@typeset@protect
3474 \Hy__hypSectionAnchor{#1}%
3475 \fi
3476 }
3477 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3478 \leavevmode
3479 \hbox to 0pt{%
3480 \kern-\Hy@SectionHShift
3481 \Hy@raisedlink{%
3482 \hyper@anchorstart{#1}\hyper@anchorend
3483 }%
3484 \hss
3485 }%
3486 }
3487 \@ifundefined{hyper@nopatch@sectioning}
3488 {
3489 \let\H@old@ssect\@ssect
3490 \def\@ssect#1#2#3#4#5{%
3491 \Hy@MakeCurrentHrefAuto{section*}%
3492 \setlength{\Hy@SectionHShift}{#1}%

```

```

3493 \begingroup
3494 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3495 \toks\tw@{\expandafter{%
3496 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}}%
3497 #5%
3498 }%
3499 \edef\x{\endgroup
3500 \the\toks@{\the\toks\tw@}%
3501 }\x
3502 }
3503 \let\H@old@schapter\@schapter
3504 \def\@schapter#1{%
3505 \begingroup
3506 \let\@mkboth\@gobbletwo
3507 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3508 \Hy@raisedlink{%
3509 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3510 }%
3511 \endgroup
3512 \H@old@schapter{#1}%
3513 }
3514 \ltx@ifundefined{@chapter}{}{-%
3515 \let\Hy@org@chapter\@chapter
3516 \def\@chapter{%
3517 \def\Hy@next{%
3518 \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3519 \Hy@raisedlink{%
3520 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3521 }%
3522 }%
3523 \ifnum\c@secnumdepth>\m@ne
3524 \ltx@ifundefined{if@mainmatter}%
3525 \iftrue{\csname if@mainmatter\endcsname}%
3526 \let\Hy@next\relax
3527 \fi
3528 \fi
3529 \Hy@next
3530 \Hy@org@chapter
3531 }%
3532 }
3533 \let\H@old@part\@part
3534 \begingroup\expandafter\expandafter\expandafter\endgroup
3535 \expandafter\ifx\csname chapter\endcsname\relax
3536 \let\Hy@secnum@part\z@
3537 \else
3538 \let\Hy@secnum@part\m@ne
3539 \fi
3540 \def\@part{%
3541 \ifnum\Hy@secnum@part>\c@secnumdepth
3542 \phantomsection
3543 \fi
3544 \H@old@part
3545 }
3546 \let\H@old@spart\@spart

```

```

3547 \def\@spart#1{%
3548   \Hy@MakeCurrentHrefAuto{part*}%
3549   \Hy@raisedlink{%
3550     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3551   }%
3552   \H@old@spart{#1}%
3553 }
3554 \let\H@old@sect\@sect
3555 \def\@sect#1#2#3#4#5#6[#7]#8{%
3556   \ifnum #2>\c@secnumdepth
3557     \expandafter\@firstoftwo
3558   \else
3559     \expandafter\@secondoftwo
3560   \fi
3561   {%
3562     \Hy@MakeCurrentHrefAuto{section*}%
3563     \setlength{\Hy@SectionHShift}{#3}%
3564     \begingroup
3565     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3566     \toks\tw@{\expandafter{%
3567       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3568       #8%
3569     }}%
3570     \edef\x{\endgroup
3571       \the\toks@{\the\toks\tw@}%
3572     }\x
3573   }{%
3574     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3575   }%
3576 }
3577 {}
3578 \expandafter\def\csname Parent-4\endcsname{}
3579 \expandafter\def\csname Parent-3\endcsname{}
3580 \expandafter\def\csname Parent-2\endcsname{}
3581 \expandafter\def\csname Parent-1\endcsname{}
3582 \expandafter\def\csname Parent0\endcsname{}
3583 \expandafter\def\csname Parent1\endcsname{}
3584 \expandafter\def\csname Parent2\endcsname{}
3585 \expandafter\def\csname Parent3\endcsname{}
3586 \expandafter\def\csname Parent4\endcsname{}
3587 %%
3588 %% End of file 'hgeneric-testphase.def'.
3589 \</package>

3590 <*colorscheme>
3591 % collected from https://tex.stackexchange.com/questions/525261/better-default-colors-for-hy
3592 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3593 % linkcolor=
3594 %,filecolor=
3595 %,urlcolor=
3596 %,menucolor=
3597 %,runcolor=
3598 %,linkbordercolor=
3599 %,filebordercolor=
3600 %,urlbordercolor=

```

```

3601 %,menubordercolor=
3602 %,runbordercolor=
3603
3604 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3605 {
3606     linkcolor      = [rgb]{1,0,0}, %red
3607     filecolor      = [rgb]{0,1,1}, %cyan
3608     urlcolor       = [rgb]{1,0,1}, %magenta
3609     menucolor      = [rgb]{1, 0, 0}, %red
3610     runcolor       = [rgb]{0,1,1}, %cyan
3611 %-----
3612     linkbordercolor = [rgb]{1, 0 ,0 },
3613     filebordercolor = [rgb]{0, .5, .5},
3614     urlbordercolor  = [rgb]{0, 1, 1},
3615     menubordercolor = [rgb]{1, 0, 0},
3616     runbordercolor  = [rgb]{0, .7, .7}
3617 }
3618
3619 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3620 {
3621     linkcolor      = [rgb]{0,0.2,0.6},
3622     filecolor      = [rgb]{0.8,0,0.8},
3623     urlcolor       = [rgb]{0.8,0,0.8},
3624     menucolor      = [rgb]{0,0.2,0.6},
3625     runcolor       = [rgb]{0.8,0,0.8},
3626 %----- %-----
3627     linkbordercolor = [rgb]{0,0.2,0.6},
3628     filebordercolor = [rgb]{0.8,0,0.8},
3629     urlbordercolor  = [rgb]{0.8,0,0.8},
3630     menubordercolor = [rgb]{0,0.2,0.6},
3631     runbordercolor  = [rgb]{0.8,0,0.8}
3632 }
3633
3634 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3635 { %two colors: intern/extern
3636     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3637     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3638     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3639     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3640     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3641 %----- %-----
3642     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3643     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3644     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3645     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3646     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3647 }
3648
3649 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3650 { %all darkgray
3651     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3652     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3653     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3654     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },

```

```

3655     runcolor          = [rgb]{0.4 ,0.4 ,0.4 },
3656 %----- %-----
3657     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3658     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3659     urlbordercolor  = [rgb]{0.4 ,0.4 ,0.4 },
3660     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3661     runbordercolor  = [rgb]{0.4 ,0.4 ,0.4 }
3662 }
3663
3664 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3665 { %dvipsnam.def
3666     linkcolor          = [rgb]{0.06, 0.46, 1}, %NavyBlue
3667     filecolor          = [rgb]{1, 0, 0}, %Red
3668     urlcolor           = [rgb]{0.06, 0.46, 1}, %NavyBlue
3669     menucolor          = [rgb]{1, 0, 0}, %Red
3670     runcolor           = [rgb]{1, 0, 0}, %Red
3671 %----- %-----
3672     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3673     filebordercolor = [rgb]{1, 0, 0}, %Red
3674     urlbordercolor  = [rgb]{0.06, 0.46, 1}, %NavyBlue
3675     menubordercolor = [rgb]{1, 0, 0}, %Red
3676     runbordercolor  = [rgb]{1, 0, 0} %Red
3677 }
3678
3679 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3680 { %dvipsnam.def
3681     linkcolor          = [rgb]{0.72, 0, 0}, %BrickRed
3682     filecolor          = [rgb]{0, 1, 0}, %Green
3683     urlcolor           = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3684     menucolor          = [rgb]{0.06, 0.46, 1}, %NavyBlue
3685     runcolor           = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3686 %----- %-----
3687     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3688     filebordercolor = [rgb]{0, 1, 0}, %Green
3689     urlbordercolor  = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3690     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3691     runbordercolor  = [rgb]{0.64, 0.08, 0.98} %Mulberry
3692 }
3693
3694
3695 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3696 {
3697     linkcolor          = [rgb]{0.50196, 0, 0.02353},
3698     filecolor          = [rgb]{0.07451, 0.09412, 0.46667},
3699     urlcolor           = [rgb]{0.54118, 0, 0.52941},
3700     menucolor          = [rgb]{0.44706, 0.45882, 0},
3701     runcolor           = [rgb]{0.07451, 0.46667, 0.46275},
3702 %----- %-----
3703     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3704     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3705     urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3706     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3707     runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3708 }

```

```

3709
3710 \prop_const_from_keyval:Nn \c_hyp_colorscheme_henryford_prop
3711 {
3712   linkcolor      = [rgb]{0,0,0},
3713   filecolor      = [rgb]{0,0,0},
3714   urlcolor       = [rgb]{0,0,0},
3715   menucolor      = [rgb]{0,0,0},
3716   runcolor       = [rgb]{0,0,0},
3717   %----- %-----
3718   linkbordercolor = [rgb]{0,0,0},
3719   filebordercolor = [rgb]{0,0,0},
3720   urlbordercolor  = [rgb]{0,0,0},
3721   menubordercolor = [rgb]{0,0,0},
3722   runbordercolor  = [rgb]{0,0,0}
3723 }
3724 </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	
\#	278, 732
\\$	277
\%	733
\.	542, 546, 548
@curropt commands:	
\@curropt:	2631, 2698
\ \	
...	19, 20, 30, 31, 32, 40, 44, 54, 74,
81, 88, 95, 102, 117, 122, 123, 131,	
132, 141, 142, 143, 144, 145, 152,	
159, 279, 280, 862, 2633, 2700, 3040	
_	540, 542, 546, 548
A	
\A	540
\Acrobatmenu	18, 172
\addcontentsline	13
\AddToDocumentProperties	387
\AddToHook	425, 438
\AddToHookNext	197
\advance	2564, 2638, 2682, 2683
allcolors (hypersetup key)	975
\author	2
B	
\b	544
\begingroup	208,
276, 374, 2470, 2598, 2627, 2639,	
2673, 3062, 3493, 3505, 3534, 3564	
\belowpdfbookmark	3455
\bgroup	273, 374
\BOOKMARK	3458
bookmarkstyle (hypersetup key)	13
bool commands:	
\bool_if:N	266,
295, 354, 679, 695, 707, 720, 728,	
758, 816, 873, 945, 955, 1199, 1210	
\bool_if:nTF	828, 1881, 1906, 3199, 3307
\bool_lazy_and:nnTF	422
\bool_lazy_or:nnTF	1229, 1278
\bool_new:N	202, 203, 526, 530, 534
\bool_set_true:N	535
bordercolormodel (hypersetup key)	13, 996
box commands:	
\box_dp:N	616, 2783, 2834, 2869
\box_ht:N	615, 2782, 2833, 2868
\box_new:N	475, 537
\box_set_dp:Nn	2078
\box_set_ht:Nn	2085
\box_use:N	1216, 2785, 2843, 2872
\box_use_drop:N	1221
\box_wd:N	614, 2781, 2832, 2867
\l_tmpa_box	2779, 2781, 2782,
2783, 2785, 2818, 2832, 2833, 2834,	
2843, 2865, 2867, 2868, 2869, 2872	
C	
\catcode	277, 278
\char	2717, 2914, 2926
\chardef	157
\cite	33
clist commands:	
\clist_map_function:nN	124, 133
\clist_map_inline:nn	2257
\clist_map_variable:nNn	2630, 2697
color commands:	
\color_export:nnN	38, 430, 904, 1017
\color_select:n	912, 948, 1220, 2824, 2828
\color_select:nn	918
\color_set:nn	2, 392, 929
\color_set:nnn	2, 391, 935
color names:	
hyp/annot/file	523
hyp/annot/link	523
hyp/annot/menu	523
hyp/annot/run	523
hyp/annot/url	523
colorfile (hypersetup key)	975
colorlink (hypersetup key)	975
colorlinks (hypersetup key)	961
colormenu (hypersetup key)	975
colorrn (hypersetup key)	975
colorscheme (hypersetup key)	1, 1387
colorurl (hypersetup key)	975
cs commands:	
\cs_generate_variant:Nn	155, 156,
389, 454, 469, 474, 578, 904, 921, 938	
\cs_gset:Npn	1191
\cs_gset_eq:NN	408, 419
\cs_if_exist:N	12
\cs_if_exist_p:N	423
\cs_new:Npn	117, 470
\cs_new_protected:Npn	208,
304, 321, 338, 364, 384, 399, 410,	
448, 457, 554, 558, 562, 566, 579,	

581, 585, 629, 637, 646, 650, 672, 677, 693, 705, 718, 756, 814, 871, 905, 916, 922, 933, 1135, 2113, 2184	\DocumentMetadata
\cs_set:Nn 214, 256	. . . 3, 10, 14, 15, 18, 12, 20, 32, 41, 47
\cs_set_eq:NN	draft (hypersetup key) 1411
. 160, 165, 166, 732, 733, 1445	
\cs_set_protected:Npn	E
. 158, 179, 595, 862	\edef 2472, 2641, 2677, 2684, 3025, 3499, 3570
\csname 2645, 3032, 3043, 3525, 3535, 3578, 3579, 3580, 3581, 3582, 3583, 3584, 3585, 3586	\EdefEscapeName 2696, 2701
\currentpdfbookmark 3451	\egroup 377
	\else 2469, 2594, 2648, 2652, 2655, 2672, 2680, 3013, 3018, 3035, 3041, 3053, 3057, 3074, 3078, 3082, 3086, 3090, 3094, 3098, 3102, 3106, 3110, 3127, 3131, 3232, 3233, 3234, 3236, 3239, 3243, 3247, 3253, 3270, 3274, 3278, 3284, 3288, 3356, 3357, 3359, 3362, 3366, 3391, 3392, 3394, 3397, 3401, 3468, 3537, 3558
D	\endcsname 2659, 3009, 3032, 3043, 3525, 3535, 3578, 3579, 3580, 3581, 3582, 3583, 3584, 3585, 3586
\d 542, 546, 548	\endgroup 282, 285, 375, 377, 2473, 2618, 2641, 2687, 2690, 3069, 3499, 3511, 3534, 3570
debug (hypersetup key) 1411	\endinput 3469
\DeclareRobustCommand	exp commands:
. . . 261, 289, 312, 329, 348, 377, 3477	\exp_args:Ne 3117
\def 164, 374, 375, 377, 590, 2433, 2437, 2444, 2462, 2491, 2504, 2562, 2571, 2586, 2587, 2590, 2591, 2604, 2605, 2621, 2622, 2625, 2626, 2664, 2693, 2745, 2759, 2760, 2772, 2800, 2802, 2803, 2813, 2848, 2850, 2851, 2862, 2881, 2883, 2884, 2886, 2887, 2896, 2943, 3010, 3020, 3022, 3027, 3039, 3050, 3061, 3072, 3115, 3124, 3136, 3189, 3223, 3258, 3297, 3347, 3382, 3419, 3426, 3433, 3440, 3472, 3490, 3504, 3516, 3517, 3540, 3547, 3555, 3578, 3579, 3580, 3581, 3582, 3583, 3584, 3585, 3586	\exp_args:Nne 2222
\DefaultHeightofCheckBox 2887	\exp_args:Nnno 323, 340
\DefaultHeightofChoiceMenu 2626	\exp_args:Nno 306, 366, 2170
\DefaultHeightofReset 2851	\exp_args:Nnox 766
\DefaultHeightofSubmit 2803	\exp_args:NNV 1866, 1897
\DefaultHeightofText 2595	\exp_args:NnV 631, 640
\DefaultHeightofTextMultiline 2593	\exp_args:Nnx 2310, 2415
\DefaultOptionsofCheckBox 2890	\exp_args:No
\DefaultOptionsofPushButton 2764 664, 668, 2762, 2805, 2853, 2888
\DefaultOptionsofReset 2855	\exp_args:NV 612, 619
\DefaultOptionsofSubmit 2807	\exp_args:Nx 762, 2163, 2211, 2464
\DefaultOptionsofText 2600	\exp_not:N 862
\DefaultWidthofCheckBox 2886	\exp_not:n
\DefaultWidthofChoiceMenu 2625	. . . 1326, 1360, 1452, 1506, 1527, 1564, 1584, 1607, 1651, 1684, 1709, 1760, 1785, 1812, 1861, 1938, 1966, 1993, 2013, 2243, 2360, 2388, 2398, 2409
\DefaultWidthofReset 2850	\expandafter 279, 2599, 2633, 2642, 2644, 2645, 2700, 3026, 3042, 3469, 3495, 3496, 3534, 3535, 3557, 3559, 3566, 3567, 3578, 3579, 3580, 3581, 3582, 3583, 3584, 3585, 3586
\DefaultWidthofSubmit 2802	\ExplSyntaxOff
\DefaultWidthofText 2590	. . . 9, 2501, 2554, 2581, 2939, 3007, 3123, 3187, 3222, 3294, 3345, 3444
destlabel (hypersetup key) 13	\ExplSyntaxOn
dim commands:	. . . 7, 10, 2502, 2559, 2582, 2942, 3114, 3135, 3188, 3257, 3295, 3381
\dim_to_decimal_in_bp:n 11, 160	
\do 2631, 2698	
\documentclass 21, 33	

extension (hypersetup key)	13, 1429	hidemenu (hypersetup key)	1363
F			
\fbbox	2736, 2824, 2828	hiderun (hypersetup key)	1363
\fi	2458,	hideurl (hypersetup key)	1363
	2485, 2488, 2497, 2596, 2604, 2605,	hook commands:	
	2636, 2654, 2657, 2658, 2664, 2667,	\hook_gput_code:nnn	
	2676, 2678, 2685, 2688, 3015, 3037,	401, 412, 941, 951, 1195, 1206
	3047, 3049, 3055, 3059, 3066, 3067,	\hook_new:n	390, 553, 628
	3076, 3080, 3084, 3088, 3092, 3096,	\hook_new_pair:nn	
	3100, 3104, 3108, 3112, 3129, 3133,	259, 287, 311, 328, 346
	3232, 3233, 3234, 3241, 3245, 3249,	\hook_use:n	
	3251, 3253, 3272, 3276, 3280, 3284,	263, 283, 292, 308, 315, 325, 332,
	3291, 3356, 3357, 3364, 3368, 3370,		342, 351, 370, 570, 634, 643, 684, 700
	3391, 3392, 3399, 3403, 3405, 3470,	\href	4, 5, 19, 261
	3475, 3527, 3528, 3539, 3543, 3560	\hreflaunch	5
file (hypersetup key)	10, 1460	\hrefpdf	5, 19, 312, 382
file commands:		\hrefrun	5, 329, 383
\file_input:n	8	\hrefurl	5, 19, 289, 381
fileborderstyle (hypersetup key)	14, 1089	\hss	3484
filecolor (hypersetup key)	975	Hy internal commands:	
final (hypersetup key)	1411	\Hy_hyescapeform	3026, 3039, 3045
\fontencoding	2713, 2910, 2922	\Hy_hypSectionAnchor	3474, 3477
\fontfamily	2712, 2909, 2921	HyField internal commands:	
\fontseries	2714, 2911, 2923	\HyField_hypAddToFields	2464, 2491
\fontshape	2715, 2912, 2924	hyp commands:	
fp commands:		\l_hyp_annot_colorfile_bool	524
\fp_eval:n	2036, 2064, 2080	\l_hyp_annot_colorlink_bool	524
G		\l_hyp_annot_colormenu_bool	524
\gdef	279, 280	\l_hyp_annot_colorrrun_bool	524
\GetDocumentProperties	3	\l_hyp_annot_colorurl_bool	524
\global	2564, 3005	\l_hyp_annot_ocgcolorfile_bool	528
group commands:		\l_hyp_annot_ocgcolorlink_bool	528
\group_begin:	264, 293, 316, 333,	\l_hyp_annot_ocgcolormenu_bool	528
	352, 568, 682, 698, 722, 760, 818,	\l_hyp_annot_ocgcolorrrun_bool	528
	875, 947, 1202, 1219, 2339, 2711,	\l_hyp_annot_ocgcolorurl_bool	528
	2761, 2804, 2852, 2885, 2908, 2920	\l_hyp_annot_ocgcolorurl_bool	528
\group_end:	307, 324, 341,	\l_hyp_current_dest_name_tl	
	369, 575, 687, 710, 746, 806, 856,	627, 683, 699
	899, 957, 1222, 1225, 2346, 2718,	hyp internal commands:	
	2797, 2845, 2878, 2915, 2927, 2937	\g__hyp_AcroForm_CoFields_prop	
H		2425, 2446, 2515, 2517, 2532
\hbox	374, 3479	\g__hyp_AcroForm_Fields_prop	
hbox commands:		2426, 2439, 2510
\hbox_overlap_right:n	1216	\l__hyp_annot_GoTo_bool	679, 695, 707
\hbox_set:Nn	2779, 2818, 2865	\l__hyp_annot_GoTo_bool\l__-	
\hbox_set:Nw	1203	_hyp_annot_URI_bool\l__-	
\hbox_set_end:	1212	hyp_annot_GoToR_bool\l__-	
\hbox_set_to_wd:Nnn	2062	hyp_annot_Named_bool\l__-	
hidefile (hypersetup key)	1363	hyp_annot_Launch_bool	532
hidelink (hypersetup key)	1363	\l__hyp_annot_GoToR_bool	758
hidelinks (hypersetup key)	1363	\l__hyp_annot_Launch_bool	816
		\l__hyp_annot_Named_bool	873
		\c__hyp_annot_types_seq	
		487, 524, 528, 975, 1276, 1374
		\l__hyp_annot_URI_bool	720

\g__hyp_bordercolormodel_str . . .	__hyp_href_url_aux:nn 302, 304
..... 432, 523, 1000, 1019	\l__hyp_href_url_encode_bool . . .
__hyp_citebordercolor_hook_- 202, 213, 251, 266, 295, 354
init: 397, 410, 419	__hyp_href_url_format:
__hyp_citecolor_hook_init: 208, 214, 256, 367
..... 395, 399, 408	\l__hyp_href_url_ismap_bool
__hyp_clist_display:n . 117, 124, 133 203, 227, 728
__hyp_color_select:n 39, 905, 905, 921	\l__hyp_href_url_protocol_tl . . .
__hyp_color_select_aux:wn 204, 215, 254, 306, 368
..... 905, 909, 916	__hyp_link_goto_begin:nw
__hyp_color_set:nn 38, 39, 650, 685, 701
395, 397, 429, 922, 922, 938, 980, 1016	__hyp_link_goto_end: . 672, 686, 709
__hyp_color_set_aux:nwn 922, 926, 933	\c__hyp_map_annot_hyp_prop 487
\c__hyp_colorscheme_daleif_prop 3619	\c__hyp_map_hyp_annot_prop
\c__hyp_colorscheme_henryford_- 487, 939,
prop 3710	1003, 1043, 1070, 1078, 1089, 1116,
\c__hyp_colorscheme_julian_prop 3634	1124, 1193, 1301, 1337, 1347, 1460
\c__hyp_colorscheme_phelype_prop	__hyp_ocg_init: 1135, 1135, 1191, 1201
..... 3695	l__hyp_page/Trans 551
\c__hyp_colorscheme_szabolcsA_-	__hyp_PageLabels_gpush: 585, 585, 598
prop 3664	\l__hyp_para_tmpa_tl 483, 827, 830, 839
\c__hyp_colorscheme_szabolcsB_-	\l__hyp_para_tmpa_tl\l__hyp_-
prop 3679	text_tmpa_str\g__hyp_text_-
\c__hyp_colorscheme_tivv_prop . 3649	tmpa_str 480
\l__hyp_dest_box 31,	__hyp_ref_check:nn 447, 457, 469, 2569
537, 614, 615, 616, 2062, 2078, 2085	__hyp_ref_if_exist:nn . 447, 455, 456
\l__hyp_dest_name_tmpa_tl . . 480,	__hyp_ref_if_exist:nnTF . . 459, 2573
653, 654, 660, 664, 666, 669, 779, 792	__hyp_ref_label:nn 447, 448, 454, 2568
\l__hyp_dest_pdfremotestartview_-	__hyp_ref_value:nn 447, 470, 474, 2577
tl 506, 787, 1869, 1873	__hyp_secondoftwoewithopt:wnn . . .
\g__hyp_dest_pdfstartpage_tl 380, 381, 382, 383
... 506, 1880, 1882, 1889, 1907, 1914	__hyp_setup_info_date_key:nn . . .
\g__hyp_dest_pdfstartview_tl 2184, 2216, 2217
506, 1882, 1889, 1900, 1904, 1907, 1914	__hyp_setup_info_key:nn . . 2113,
\l__hyp_dest_pdfview_tl	2177, 2178, 2179, 2180, 2182, 2183
..... 522, 632, 641,	__hyp_store_metadata:nn
2031, 2034, 2041, 2044, 2045, 2046, 384, 389, 2110,
2047, 2048, 2049, 2054, 2058, 2094	2145, 2163, 2197, 2211, 2232, 2304
\c__hyp_dest_startview_regex . . .	__hyp_text_cleanup:N . 558, 558, 572
..... 538, 1867, 1898	\l__hyp_text_enc_dest_print_tl . .
\c__hyp_dest_undefined_tl 509, 778
..... 486, 659, 660	\l__hyp_text_enc_dest_tl
__hyp_destination:nn 31, 509, 608, 665
..... 31, 601, 601, 632, 641	\l__hyp_text_enc_file_print_tl . .
\l__hyp_filename_tmpa_tl . . . 480, 509, 821
761, 763, 769, 770, 775, 822, 823, 839	\l__hyp_text_enc_info_print_tl . .
__hyp_href_pdf_aux:nn 318, 321 509, 583
\l__hyp_href_pdf_destination_tl . .	\l__hyp_text_enc_para_print_tl . .
..... 205, 216, 255, 323 509, 826
\l__hyp_href_pdf_page_tl 206, 224, 786	\l__hyp_text_enc_uri_print_tl . . .
__hyp_href_run_aux:nn 335, 338 268,
\l__hyp_href_run_parameter_tl . . .	271, 297, 300, 356, 359, 509, 725, 1471
..... 207, 228, 340	
__hyp_href_url_aux:n 361, 364	

_hyp_text_pdfstring:nnN	hidelink	1363
.....	566, 566, 578, 580,	hidelinks	1363
583, 606, 663, 723, 776, 819, 824, 1471		hidemenu	1363
_hyp_text_pdfstring_info:nN	...	hiderun	1363
.....	581, 581, 2139, 2157, 2167	hideurl	1363
_hyp_text_purify:nN	. 554, 554, 571	hypertextnames	1429
_hyp_text_string_from_unicode:nN	link	10, 1460
.....	562, 562, 573	linkborder	13
\g_hyp_text_tmpa_str	. 485, 574, 576	linkborderstyle	14, 1089
\l_hyp_text_tmpa_str	linkcolor	975
.....	484, 571, 572, 573, 574	linkfileprefix	1429
\l_hyp_tmpa_box	475, 1203, 1216, 1221	linktoc	1429
\l_hyp_tmpa_int	linktocpage	1429
.....	475	localanchorname	1429
\l_hyp_tmpa_seq	menu	10, 1460
.....	475, 1867,	menuborder	13
1869, 1898, 1900, 2021, 2022, 2026,		menuborderstyle	14, 1089
2028, 2052, 2059, 2060, 2061, 2070,		menucolor	975
2072, 2082, 2087, 2519, 2521, 2527		naturalnames	1429
\l_hyp_tmpa_str	nesting	13
.....	475, 2139, 2140, 2142,	ocgcolorfile	1229
2157, 2158, 2160, 2167, 2168, 2172		ocgcolorlink	1229
\l_hyp_tmpa_tl	475, 609, 613, 621,	ocgcolorlinks	1229
1020, 1024, 1471, 1472, 1477, 1865,		ocgcolormenu	1229
1867, 1896, 1898, 2028, 2029, 2036		ocgcolorrun	1229
\l_hyp_uri_tmpa_tl	... 480, 726, 727	ocgcolorurl	1229
hyp/anchor	pageanchor	1429
hyp/annot/file (color name)	pdfauthor	2113
hyp/annot/link (color name)	pdfborder	13
hyp/annot/menu (color name)	pdfborderstyle	14, 1089
hyp/annot/run (color name)	pdfcreationdate	14, 2184
hyp/annot/url (color name)	pdfcreator	2113
hyp/text/pdfstring	pdfencoding	1401
.....	553	pdfinfo	2246
\hypercalcbp	pdfkeywords	2113
.....	11, 18, 160	pdflang	14, 2099
\HyperDestNameFilter	pdflinkmargin	14
.....	13, 607, 664	pdfmoddate	14, 2184
\hypersetup	2, 3, 5, 9, 10, 13, 18, 66, 99, 179	pdfproducer	2113
\hypersetup keys:		pdfremotestartview	11
allcolors	pdfstartview	11
.....	975	pdfsubject	2113
bookmarkstyle	pdftitle	2113
.....	13	pdftrapped	2218
bordercolormodel	pdfversion	1401
.....	13, 996	pdfview	11, 2017
colorfile	plainpages	1429
.....	975	run	10, 1460
colorlink	runborder	13
.....	975	runborderstyle	14, 1089
colorlinks	runcolor	975
.....	961	unicode	1401
colormenu	url	10, 1460
.....	975	urlborder	13
colorrun			
.....	975			
colorscheme			
.....	1, 1387			
colorurl			
.....	975			
debug			
.....	1411			
destlabel			
.....	13			
draft			
.....	1411			
extension			
.....	13, 1429			
file			
.....	10, 1460			
fileborderstyle			
.....	14, 1089			
filecolor			
.....	975			
final			
.....	1411			
hidefile			
.....	1363			

urlborderstyle	14, 1089		
urlcolor	975		
verbose	1411		
hypertextnames (hypersetup key)	1429		
hypListbox internal commands:			
_hypListbox	2686, 2745		
hypRadio internal commands:			
_hypRadio	2671, 2693		
hypwritetorep internal commands:			
_hypwritetorep	3448		
I			
\ifcase	3232, 3356, 3391		
\ifcsname	3009		
\ifdim	2636, 2675, 2676		
\ifnum	3063, 3232, 3238, 3269, 3356, 3361, 3391, 3396, 3523, 3541, 3556		
\iftrue	3525		
\ifx	2468, 2605, 2665, 3040, 3052, 3056, 3073, 3077, 3081, 3085, 3089, 3093, 3097, 3101, 3105, 3109, 3126, 3130, 3233, 3234, 3242, 3246, 3253, 3273, 3277, 3284, 3285, 3357, 3365, 3392, 3400, 3473, 3535		
\immediate	2452, 2455		
int commands:			
\int_compare:nNnTF	2026, 2052, 2737, 3155, 3208, 3318, 3340		
\int_compare_p:nNn	3201, 3309		
\int_eval:n	785		
\int_max:nn	786		
\int_new:N	478		
iow commands:			
\iow_newline:	2958, 2964, 2977		
K			
\kern	3480		
keys commands:			
\keys_define:nn	185, 193, 211, 249, 393, 961, 977, 984, 996, 1005, 1030, 1045, 1064, 1091, 1110, 1233, 1247, 1259, 1282, 1288, 1303, 1332, 1363, 1376, 1387, 1401, 1411, 1416, 1429, 1441, 1462, 1467, 2017, 2099, 2115, 2148, 2186, 2200, 2218, 2246, 2302, 2317, 2330, 2350		
\l_keys_key_str	190, 2172, 2415		
\keys_set:nn	182, 219, 238, 265, 294, 317, 334, 353, 403, 414, 1393, 1400, 2223, 2250, 2253, 2254, 2255, 2256, 2311, 2340, 2415		
\keys_set_known:nn	2418		
\kvsetkeys	181, 2506		
L			
\label	9, 13		
\LayoutCheckField	2898		
\LayoutChoiceField	2668		
\LayoutPushButtonField	2774, 2791		
\LayoutTextField	2606		
\leavevmode	2607, 2704, 3478		
legacy commands:			
\legacy_if:nTF	167, 436, 443, 593, 2544, 2770, 2811, 2861, 2894, 3178		
\let	381, 382, 383, 690, 741, 748, 802, 808, 852, 858, 901, 2428, 2429, 2431, 2432, 2459, 2555, 2556, 2557, 2588, 2589, 2623, 2624, 2666, 3005, 3012, 3014, 3019, 3036, 3070, 3489, 3503, 3506, 3515, 3526, 3533, 3536, 3538, 3546, 3554		
link (hypersetup key)	10, 1460		
linkborder (hypersetup key)	13		
linkborderstyle (hypersetup key)	14, 1089		
linkcolor (hypersetup key)	975		
linkfileprefix (hypersetup key)	1429		
linktoc (hypersetup key)	1429		
linktocpage (hypersetup key)	1429		
localanchorname (hypersetup key)	1429		
M			
\MakeButtonField	2779, 2794, 2818, 2865, 2876		
\MakeChoiceField	2754		
\MakeFieldObject	2419		
\MakeRadioField	2736		
\MakeTextField	2615		
\mbox	1213		
menu (hypersetup key)	10, 1460		
menuborder (hypersetup key)	13		
menuborderstyle (hypersetup key)	14, 1089		
menucolor (hypersetup key)	975		
mode commands:			
\mode_if_horizontal:TF	603, 625		
\mode_leave_vertical:	262, 291, 314, 331, 350, 652, 735, 794, 841, 878, 2748, 2776, 2793, 2815, 2857, 2900		
msg commands:			
\msg_error:nn	424, 2790, 2875		
\msg_info:nnn	2125, 2130		
\msg_line_context:	82		
\g_msg_module_name_prop	11		
\msg_new:nnn	50, 57, 62, 66, 70, 77, 84, 91, 98, 105, 111, 118, 127, 137, 148		
\msg_new:nnnn	14, 25, 36		
\msg_warning:nn	169, 1407		
\msg_warning:nnn	178, 656, 896		

`\msg_warning:nnnn` 1251, 1292, 1504, 1548,
1582, 1605, 1667, 1696, 1730, 1745,
1758, 1771, 1796, 1829, 1845, 1872,
1903, 1936, 1950, 1977, 2011, 2093
`\msg_warning:nnnnn` 189,
1323, 1357, 1449, 1524, 1561, 1648,
1681, 1706, 1782, 1809, 1858, 1963,
1990, 2240, 2357, 2376, 2395, 2406

N

`naturalnames` (hypersetup key) [1429](#)
`nesting` (hypersetup key) [13](#)
`\newcommand` [163](#), [3452](#), [3454](#), [3456](#)
`\newcount` [2560](#), [2744](#)
`\NewDocumentCommand` [2419](#)
`\NewExpandableDocumentCommand` [380](#)
`\newlength` [3471](#)
`\noexpand` ... [2642](#), [2643](#), [2644](#), [2645](#), [3023](#)
`\nolinkurl` [4](#)

O

`ocgcolorfile` (hypersetup key) [1229](#)
`ocgcolorlink` (hypersetup key) [1229](#)
`ocgcolorlinks` (hypersetup key) [1229](#)
`ocgcolormenu` (hypersetup key) [1229](#)
`ocgcolorrun` (hypersetup key) [1229](#)
`ocgcolorurl` (hypersetup key) [1229](#)

P

`pageanchor` (hypersetup key) [1429](#)
`\paperwidth` [3](#)
`\PassOptionsToPackage` ... [445](#), [1421](#), [1426](#)
pdf commands:
`\pdf_bdcobject:nn` [1215](#), [1218](#)
`\pdf_destination:nn` [27](#), [155](#), [620](#)
`\pdf_destination:nnnn` [613](#)
`\pdf_emc:` [1217](#), [1223](#)
`\pdf_link_user:nnn` [2751](#)
`\pdf_name_from_unicode_e:n`
..... [244](#), [765](#), [880](#)
`\pdf_object_if_exist:nTF` [763](#)
`\pdf_object_new:nn` [1137](#),
[1138](#), [1139](#), [1140](#), [2945](#), [2946](#), [2947](#)
`\pdf_object_ref:n` [156](#), [775](#),
[1143](#), [1145](#), [1167](#), [1168](#), [1171](#), [1175](#),
[1180](#), [1185](#), [1190](#), [2537](#), [2539](#), [3003](#)
`\pdf_object_ref_last:` [796](#), [2345](#)
`\pdf_object_unnamed_write:nn` ...
..... [795](#), [2341](#)
`\pdf_object_write:nn` [1141](#),
[1147](#), [1157](#), [1169](#), [2948](#), [2990](#), [2997](#)
`\pdf_pageobject_ref:n` [1889](#), [1914](#), [2575](#)
`\pdf_string_from_unicode:nnN` .. [564](#)

`\pdf_version:`
... [1552](#), [1671](#), [1700](#), [1734](#), [1749](#),
[1775](#), [1800](#), [1833](#), [1849](#), [1954](#), [1981](#)
`\pdf_version_compare:NnTF`
[1542](#), [1633](#), [1655](#), [1691](#), [1724](#), [1739](#),
[1765](#), [1790](#), [1816](#), [1839](#), [1944](#), [1971](#)
`\pdf_version_compare_p:Nn`
..... [832](#), [1230](#), [1279](#)
`\pdf_version_major:`
..... [166](#), [1231](#), [1254](#), [1280](#), [1296](#)
`\pdf_version_minor:` .. [165](#), [1254](#), [1296](#)
pdfannot commands:
`\pdfannot_box:nnnn`
... [2610](#), [2721](#), [2780](#), [2831](#), [2866](#), [2930](#)
`\pdfannot_box_ref_last:` .. [2466](#), [2482](#)
`\pdfannot_dict_put:nnn`
..... [736](#), [796](#), [842](#), [881](#), [1021](#),
[1056](#), [1080](#), [1102](#), [1126](#), [1308](#), [1339](#)
`\pdfannot_dict_remove:nn` .. [1011](#),
[1051](#), [1072](#), [1097](#), [1118](#), [1316](#), [1349](#)
`\pdfannot_link:nnn` . [737](#), [797](#), [843](#), [882](#)
`\pdfannot_link_goto_begin:nw` .. [669](#)
`\pdfannot_link_goto_end:` [674](#)
`\pdfannot_link_margin:n` [8](#), [1586](#)
`\c_pdfannot_link_types_seq` [532](#)
pdfauthor (hypersetup key) [2113](#)
\pdfbookmark [3457](#)
pdfborder (hypersetup key) [13](#)
pdfborderstyle (hypersetup key) . [14](#), [1089](#)
pdfcreationdate (hypersetup key) [14](#), [2184](#)
pdfcreator (hypersetup key) [2113](#)
\pdfdest [11](#)
pdfdict commands:
`\pdfdict_new:n` [551](#), [714](#), [751](#), [810](#), [867](#)
`\pdfdict_put:nnn`
... [236](#), [243](#), [552](#), [715](#), [716](#), [727](#),
[730](#), [752](#), [753](#), [765](#), [772](#), [782](#), [792](#),
[811](#), [812](#), [823](#), [838](#), [868](#), [869](#), [879](#),
[1612](#), [1613](#), [1617](#), [1618](#), [2354](#), [2364](#),
[2368](#), [2371](#), [2373](#), [2392](#), [2401](#), [2403](#)
`\pdfdict_remove:nn` [233](#), [835](#), [1622](#), [1623](#)
`\pdfdict_use:n`
... [736](#), [795](#), [842](#), [848](#), [881](#), [886](#), [2343](#)
pdfencoding (hypersetup key) [1401](#)
\pdfescapestring [3012](#)
pdffile commands:
`\pdffile_embed_file:nnn` [767](#)
PDFForm internal commands:
`\PDFForm_hypName`
..... [3051](#), [3054](#), [3058](#), [3061](#)
pdfinfo (hypersetup key) [2246](#)
pdfkeywords (hypersetup key) [2113](#)
pdflang (hypersetup key) [14](#), [2099](#)
pdflinkmargin (hypersetup key) [14](#)

pdfmanagement commands:	prop commands:
\pdfmanagement_add:nn 1595	\prop_const_from_keyval:Nn 489, 497, 3604, 3619, 3634, 3649, 3664, 3679, 3695, 3710
\pdfmanagement_add:nnn 587, 1167, 1168, 1190, 1477, 1496, 1512, 1516, 1536, 1544, 1573, 1628, 1638, 1662, 1676, 1688, 1693, 1718, 1726, 1741, 1767, 1792, 1824, 1841, 1887, 1912, 1926, 1946, 1973, 2002, 2108, 2126, 2131, 2142, 2160, 2171, 2195, 2209, 2227, 2326, 2345, 2512, 2529, 2536, 2538, 2540, 2546	\prop_gput:Nnn 11, 2439, 2446
\pdfmanagement_if_active:p: ... 423	\prop_if_empty:NTF 2515
\pdfmanagement_remove:nn 1474, 1492, 1500, 1520, 1532, 1557, 1569, 1577, 1591, 1600, 1644, 1659, 1678, 1715, 1754, 1779, 1805, 1820, 1854, 1884, 1909, 1922, 1931, 1959, 1986, 1998, 2007, 2105, 2135, 2154, 2192, 2206, 2323, 2336, 2550	\prop_item:Nn 2532
\pdfmanagement_show:n 2513	\prop_map_inline:Nn 939, 1003, 1043, 1070, 1078, 1089, 1116, 1124, 1193, 1301, 1337, 1347, 1391, 1460, 2510, 2517
pdfmeta commands:	\prop_new:N 2425, 2426
\pdfmeta_standard_item:n 2308, 2313, 2314	\protect 461, 690, 741, 748, 802, 808, 852, 858, 901, 3473
\pdfmeta_standard_verify:nnTF 876, 2767, 2858	\providecommand 161, 162, 2453, 2456, 2620, 3008
\pdfmeta_standard_verify:nTF 2542, 3119	\ProvidesFile 3
pdfmoddate (hypersetup key) 14, 2184	
pdfproducer (hypersetup key) 2113	
pdfremotestartview (hypersetup key) .. 11	
pdfstartview (hypersetup key) 11	
\pdfstringdef 8, 9, 22, 28, 3068	
\pdfstringdefDisableCommands 381, 382, 383	
pdfsubject (hypersetup key) 2113	
pdftitle (hypersetup key) 2113	
pdftrapped (hypersetup key) 2218	
pdfversion (hypersetup key) 1401	
pdfview (hypersetup key) 11, 2017	
pdfxform commands:	
\pdfxform_if_exist:nTF 2707, 2819, 2903	
\pdfxform_new:nnn 2421, 2709, 2822, 2826, 2905, 2917	
\pdfxform_ref:n 2731, 2732, 2838, 2839, 3150, 3151	
\phantom 2926	
\phantomsection 12, 19, 3542	
plainpages (hypersetup key) 1429	
prg commands:	
\prg_do_nothing: 408, 419	
\prg_generate_conditional_- variant:Nnn 456	
\prg_new_eq_conditional:NNn ... 455	
	R
	\ReadBookmarks 3462
	ref commands:
	\ref_if_exist:nn 455
	\ref_label:nn 451
	\ref_value:nn 25, 472
	\refstepcounter 12
	regex commands:
	\regex_const:Nn 538
	\regex_extract_once:NnNTF 1867, 1898
	\relax 690, 741, 748, 802, 808, 852, 858, 901, 2623, 2624, 2665, 3005, 3052, 3056, 3126, 3130, 3233, 3234, 3242, 3246, 3273, 3277, 3357, 3365, 3392, 3400, 3526, 3535
	\RemoveFromHook 1482
	\RenewDocumentCommand 172
	\RequirePackage 6, 31, 440, 3460
	run (hypersetup key) 10, 1460
	runborder (hypersetup key) 13
	runborderstyle (hypersetup key) . 14, 1089
	runcolor (hypersetup key) 975
	S
	\selectfont 2716, 2913, 2925
	seq commands:
	\seq_const_from_clist:Nn 487
	\seq_count:N 2026, 2052
	\seq_get_right:NN 2028
	\seq_item:Nn 1869, 1900, 2022, 2070, 2072, 2082, 2087
	\seq_map_inline:Nn 524, 528, 532, 975, 1276, 1374, 2527
	\seq_new:N 477
	\seq_put_right:Nn 2059, 2060, 2061, 2519

<code>\seq_set_split:Nnn</code>	2021	<code>\@currDisplay</code>	2635, 2703
<code>\seq_sort:Nn</code>	2521	<code>\@currValue</code>	2701, 2731
<code>\setbox</code>	374	<code>\@currentHref</code>	
<code>\setlength</code>	3492, 3563	3496, 3509, 3520, 3550, 3567
<code>\setpdflinkmargin</code>	8	<code>\@curropt</code> ..	2620, 2630, 2633, 2697, 2700
<code>\settowidth</code>	2635	<code>\@empty</code>	2589, 2605, 3073,
<code>\show</code>	2695	3077, 3081, 3085, 3089, 3093, 3097,	
sort commands:		3101, 3105, 3109, 3253, 3284, 3285	
<code>\sort_return_same:</code>	2525	<code>\@endForm</code>	2555
<code>\sort_return_swapped:</code>	2524	<code>\@esphack</code>	452
<code>\space</code>	465, 2703, 3235, 3252,	<code>\@firstofone</code> ..	2459, 3014, 3019, 3036
3253, 3283, 3284, 3358, 3390, 3393		<code>\@firstoftwo</code>	3557
<code>\spacefactor</code>	603, 625	<code>\@for</code>	2631, 2698
<code>\stockwidth</code>	3	<code>\@gobbletwo</code>	3506
str commands:		<code>\@hyper@launch</code>	37, 862
<code>\c_colon_str</code>	862	<code>\@ifnextchar</code>	273
<code>\c_hash_str</code>	732	<code>\@ifundefined</code>	3028, 3487
<code>\c_percent_str</code>	733	<code>\@latex@warning</code>	462
<code>\str_case:nnTF</code>	2121, 3415	<code>\@mainaux</code>	2452, 2455, 2475, 2494
<code>\str_case_e:nnTF</code>	2022	<code>\@mkboth</code>	3506
<code>\str_compare:nNnTF</code>	2523	<code>\@nil</code>	3026, 3039, 3045
<code>\str_gset:Nn</code>	1000	<code>\@part</code>	3533, 3540
<code>\str_gset_eq:NN</code>	574	<code>\@pdfauthor</code>	23
<code>\str_head:n</code>	2229, 2234	<code>\@pdfborder</code>	161
<code>\str_if_eq:nnTF</code>		<code>\@pdfborderstyle</code>	162
... 610, 2140, 2158, 2168, 2308, 3117		<code>\@savsf</code>	603, 625
<code>\str_if_eq_p:nn</code>	830, 1231, 1280	<code>\@schapter</code>	3503, 3504
<code>\str_lowercase:n</code>		<code>\@secondoftwo</code>	3559
... 2022, 2163, 2211, 2230, 2235		<code>\@sect</code>	3554, 3555
<code>\str_new:N</code>	479, 484, 485, 523	<code>\@spart</code>	3546, 3547
<code>\str_set:Nn</code>	556	<code>\@ssect</code>	3489, 3490
<code>\str_set_convert:Nnnn</code>	8	<code>\@tempdima</code> ..	2629, 2636, 2638, 2675,
<code>\str_set_eq:NN</code>	576	2676, 2677, 2681, 2682, 2683, 2684	
<code>\str_tail:n</code>	2230, 2235	<code>\@tempdimb</code>	2635, 2636
<code>\str_uppercase:n</code>	2223, 2229, 2234	<code>\@typeset@protect</code>	3473
<code>\string</code>	2453,	<code>\BKM@color</code>	433
2456, 2477, 2495, 3028, 3032, 3043		<code>\c@secnumdepth</code>	3523, 3541, 3556
<code>\subpdfbookmark</code>	3453	<code>\calc@bm@number</code>	3465
sys commands:		<code>\check@bm@number</code>	3464
<code>\c_sys_backend_str</code>	3415	<code>\define@key</code>	427
T			
T _E X and L ^A T _E X 2 _ε commands:			
<code>\@BOOKMARK</code>	3459	<code>\Fld@additionalactions</code> ..	3115, 3185,
<code>\@CheckBox</code>	2881	3219, 3255, 3292, 3339, 3379, 3408	
<code>\@ChoiceMenu</code>	2621	<code>\Fld@align</code>	3144, 3230, 3305
<code>\@Form</code>	2504	<code>\Fld@altname</code> ..	3052, 3054, 3126, 3128
<code>\@PushButton</code>	2759	<code>\Fld@annotflags</code>	3139,
<code>\@Reset</code>	2848	3192, 3225, 3261, 3300, 3349, 3384	
<code>\@Submit</code>	2800	<code>\Fld@annotnames</code>	3124, 3141,
<code>\@TextField</code>	2586	3194, 3227, 3263, 3302, 3351, 3386	
<code>\@bookmarksopenlevel</code>	1487	<code>\Fld@bcolor</code>	3163, 3165, 3246,
<code>\@bsphack</code>	450	3248, 3277, 3279, 3313, 3326, 3328	
<code>\@chapter</code>	3515, 3516	<code>\fld@bcolor</code>	3234
		<code>\Fld@bordercolor</code>	3159,
		3161, 3203, 3212, 3214, 3233, 3242,	

3244, 3273, 3275, 3311, 3322, 3324, 3357, 3365, 3367, 3392, 3400, 3402	\Fld@X@additionalactions
\Fld@borderstyle 3145, 3198, 3231, 3267, 3306, 3355, 3406 3072, 3117, 3120
\Fld@borderwidth . 2682, 2683, 3145, 3198, 3231, 3267, 3306, 3355, 3406	\Form@action 3375
\Fld@calculate@code . 2468, 3085, 3087	\G@refundefinedtrue 461
\Fld@calculate@sortkey 2479	\H@old@part 3533, 3544
\Fld@cbsymbol 3167	\H@old@schapter 3503, 3512
\Fld@charsize	\H@old@ssect 3554, 3565, 3574
.. 2681, 3171, 3252, 3283, 3334, 3390	\H@old@spart 3546, 3552
\Fld@checkequals 2633, 2700	\H@old@ssect 3489, 3494
\Fld@choices 3254	\href@ 273, 279
\Fld@color 3172, 3174, 3253, 3284, 3335	\href@split 279, 280
\Fld@default	\Hy@abspage 597
... 2588, 2605, 2623, 2666, 2695, 2696, 2884, 3285, 3289, 3290, 3337	\Hy@activeanchorfalse 648
\Fld@flags 3143, 3196, 3229, 3265, 3304, 3353, 3388	\Hy@activeanchortrue 639
\Fld@format@code 3077, 3079	\Hy@AtBeginDocument 2433, 2449
\Fld@height	\Hy@bookmarkstype 1488
2591, 2612, 2615, 2626, 2684, 2723, 2736, 2754, 2803, 2851, 2887, 2932	\Hy@chapapp 3507, 3518
\Fld@keystroke@code 3073, 3075	\Hy@colorlink 34
\Fld@listcount 2694, 2702, 2737, 2744	\Hy@currentbookmarklevel 3446
\Fld@mappingname 3056, 3058, 3130, 3132	\Hy@DisableOption 171
\Fld@maxlen 3340, 3342	\Hy@drafttrue 1420
\Fld@menulength 2628, 2634, 2681	\Hy@escapeform
\Fld@name 2609, 2706, 2750, 2778, 2817, 2864, 2902, 3010, 3017, 3020
.. 2587, 2622, 2760, 2883, 3051, 3125	\Hy@escapestring 3012, 3014, 3019, 3022, 3023, 3026, 3027, 3036, 3039, 3075, 3079, 3083, 3087, 3091, 3095, 3099, 3103, 3107, 3111, 3167, 3218, 3281, 3337, 3338, 3375
\Fld@onblur@code 3093, 3095	\Hy@finaltrue 1425
\Fld@onclick@code 3218	\Hy@FormObjects 2509, 2943, 3005
\Fld@onenter@code 3105, 3107	\Hy@gttemp 3068, 3070
\Fld@onexit@code 3109, 3111	\Hy@href 273
\Fld@onfocus@code 3089, 3091	\Hy@href@nextactionraw 237
\Fld@onmousedown@code ... 3097, 3099	\Hy@href@page 225
\Fld@onmouseup@code 3101, 3103	\Hy@linkfileprefix 1434
\Fld@pageobjref	\Hy@linktoc 1445
..... 2557, 2571, 3008, 3140, 3193, 3226, 3262, 3301, 3350, 3385	\Hy@MakeCurrentHref 198
\Fld@radiosymbol 3281	\Hy@MakeCurrentHrefAuto
\Fld@rotation 3155, 3157, 3201, 3208, 3210, 3232, 3238, 3240, 3269, 3271, 3309, 3318, 3320, 3356, 3361, 3363, 3391, 3396, 3398 3491, 3507, 3518, 3548, 3562
\Fld@submitflags 3377	\Hy@next 3517, 3526, 3529
\Fld@validate@code 3081, 3083	\Hy@numberline 164, 3447
\Fld@value	\Hy@org@chapter 3515, 3530
.. 2589, 2605, 2624, 2665, 2666, 3338	\Hy@OutlineName 3463
\Fld@width 2590, 2604, 2611, 2615, 2625, 2664, 2675, 2677, 2722, 2736, 2754, 2772, 2802, 2813, 2850, 2862, 2886, 2896, 2931	\Hy@OutlineRerunCheck 3461
	\Hy@pdfmajorversion 166
	\Hy@pdfminorversion 165
	\Hy@pdfstringtrue 29, 569
	\Hy@pdfversion 3063
	\Hy@pstringdef 579, 3043
	\Hy@PutCatalog 585
	\Hy@raisedlink 3481, 3508, 3519, 3549
	\Hy@RestoreLastskip 624
	\Hy@safe@activetrue 734, 2471

\Hy@SaveLastskip	604	\hyper@normalise	
\Hy@secnum@part	3536, 3538, 3541	273, 302, 318, 335, 361, 374
\Hy@SectionAnchorHref	3472, 3496, 3567	\HyPL@Labels	587, 597
\Hy@SectionHShift		\HyPL@storePageLabel	585
.....	3471, 3480, 3492, 3563	\HyPsd@SanitizeForOutFile	3450
\Hy@StepCount	2634, 2702	\if@filesw	2451, 2474, 2493
\Hy@temp	2472, 2487, 3025, 3026	\ifFld@combo	2649, 2679
\Hy@unicodedefalse	3065	\ifFld@hidden	2604, 2664
\Hy@VerboseAnchor	605	\ifFld@multiline	2592
\Hy@VerboseLinkStart	681, 697	\ifFld@popdown	2650
\Hy@VerboseLinkStop		\ifFld@radio	2646, 2669
.....	688, 711, 744, 804, 854, 892	\ifHy@implicit	3467
\Hy@VersionChecked	157	\ifHy@pdfescapeform	3011, 3021
\Hy@WrapperDef	601	\ifHy@unicode	3064
\Hy@xspace@end	685, 743, 803, 853, 891	\kv@set@family@handler	177
\HyAnn@AbsPageLabel		\ltx@empty	2428,
.....	2556, 2562, 2608,	2429, 2468, 2555, 2556, 2557, 2588
.....	2705, 2749, 2777, 2816, 2863, 2901	\ltx@gobble	2431
\HyAnn@Count	2560, 2561, 2564, 2565,	\ltx@gobbletwo	2432
.....	2566, 2567, 2568, 2569, 2573, 2577	\ltx@ifUndefined	3514, 3524
\HyField@AddToFields .	2462, 2616,	\ltx@one	2564
.....	2738, 2755, 2786, 2842, 2871, 2935	\ltx@ReturnAfterFi	3044
\HyField@afields	2428	\ltx@zero	2561
\HyField@AfterAuxOpen	2433, 2459, 2492	\m@ne	3523, 3538
\HyField@AuxAddToCoFields		\OBJ@OCG@view	163
.....	2432, 2444, 2456, 2477	\p@	2638
\HyField@AuxAddToFields		\pdf@ifdraftmode	2507
.....	2431, 2437, 2453, 2495	\PDF@SetupDoc	158
\HyField@cofields	2429	\PDFForm@Check	2902, 2934, 3136
\HyField@FlagsCheckBox	2893	\PDFForm@List	2750, 2753, 3223
\HyField@FlagsChoice	2674	\PDFForm@Name	
\HyField@FlagsPushButton	2602, 2663, 2766, 2892, 3050
.....	2769, 2809, 2860	\PDFForm@Push	2778, 2784, 3189
\HyField@FlagsRadioButton	2670	\PDFForm@Radio	2706, 2726, 3258
\HyField@FlagsSubmit	2810	\PDFForm@Reset	2864, 2870, 3382
\HyField@FlagsText	2603	\PDFForm@Submit	2817, 2836, 3347
\HyField@PDFChoices	2747	\PDFForm@Text	2609, 2614, 3297
\HyField@SetKeys	2599, 2640,	\protected@edef	653
.....	2643, 2662, 2762, 2805, 2853, 2888	\strip@pt .	3171, 3252, 3283, 3334, 3390
\HyPat@ObjRef .	3419, 3426, 3433, 3440	\toks@	3494, 3500, 3565, 3571
\hyper@@link	281	\tw@	3495, 3500, 3566, 3571
\hyper@anchor	627	\url@	374, 377
\hyper@anchorend		\Url@def	375
.....	627, 3482, 3509, 3520, 3550	\Url@HyperHook	375
\hyper@anchorstart		\XR@ext	1431
.....	627, 3482, 3509, 3520, 3550	\z@	374, 2629, 3232, 3238,
\hyper@link	33, 34, 677	3269, 3356, 3361, 3391, 3396, 3536
\hyper@linkend	33, 34, 705	\zref@labelbylist	2566
\hyper@linkfile	323, 756	\zref@labelbyprops	2565
\hyper@linklaunch ..	36, 340, 814, 865	\zref@refused	2567
\hyper@linknamed	38, 174, 871	\texorpdfstring	29
\hyper@linkstart	33, 34, 693	text commands:	
\hyper@linkurl	306, 367, 718	\text_expand:n	761
		\text_purify:n	8, 556

<code>\textsf</code>	2824, 2828	518, 519, 520, 521, 633, 642, 683,		
<code>\the</code>	597,	699, 761, 1865, 1869, 1873, 1896,		
	2565, 2566, 2567, 2568, 2569, 2573,	2031, 2034, 2041, 2044, 2045, 2046,		
	2577, 2677, 2681, 2684, 3500, 3571	2047, 2048, 2049, 2054, 2058, 2094		
<code>\theH...</code>	12	<code>\tl_set_eq:NN</code>	660	
<code>\thepage</code>	465	<code>\tl_to_str:N</code>	763, 770, 775	
<code>\ThisShouldNotHappen</code>	3030	<code>\tl_to_str:n</code>	20, 21, 31, 32, 33, 45, 464	
<code>\title</code>	2	token commands:		
tl commands:		<code>\token_to_str:N</code>	41	
<code>\c_space_tl</code>	164, 1144,	<code>\toks</code>	3495, 3500, 3566, 3571	
	1175, 1180, 1185, 2731, 2732, 2739,			
	2954, 2956, 2966, 2968, 2970, 2973,			
	2975, 3171, 3174, 3306, 3334, 3335			
<code>\tl_const:Nn</code>	486	U		
<code>\tl_gput_right:Nn</code>	597	unicode (hypersetup key)	1401	
<code>\tl_gset:Nn</code>	1880, 1900, 1904	<code>\Url</code>	4, 5, 208	
<code>\tl_if_blank:nTF</code>		url (hypersetup key)	10, 1460	
	780, 2119, 2152, 2190, 2204, 2321, 2334	<code>\url</code>	4–6, 22, 348, 374	
<code>\tl_if_empty:nTF</code>		urlborder (hypersetup key)	13	
	654, 1472, 3159, 3163, 3172, 3335	urlborderstyle (hypersetup key) .	14, 1089	
<code>\tl_if_empty:nTF</code>		urlcolor (hypersetup key)	975	
	231, 1009, 1049, 1068,	<code>\urldef</code>	5, 22, 374, 376	
	1095, 1114, 1657, 1713, 1818, 2103	<code>\Urlfont</code>	6	
<code>\tl_if_empty_p:N</code>	1882, 1907	<code>\UrlSpecial</code>	4	
<code>\tl_if_eq:NnTF</code>	2029	use commands:		
<code>\tl_if_exist:nTF</code>	3212, 3322, 3326	<code>\use:n</code>	860	
<code>\tl_if_exist_p:N</code>	3203, 3311, 3313			
<code>\tl_if_head_eq_charcode:nNTF</code>	907, 924	V		
<code>\tl_new:N</code> . .	204, 205, 206, 207, 476,	verbose (hypersetup key)		1411
	480, 481, 482, 483, 506, 507, 508,			
	509, 510, 511, 512, 513, 514, 522, 627			
<code>\tl_set:Nn</code> .	224, 225, 237, 268, 271,	W		
	297, 300, 356, 359, 386, 516, 517,	<code>\write</code>		2452, 2455, 2475, 2494
		X		
		<code>\x</code>	2641, 2661, 3499, 3501, 3570, 3572	