

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.96s, released 2025-06-23

This module generates a generic driver `hgeneric.def/hgeneric-testphase.def` for `hyperref` meant to be used with the new L^AT_EX PDF management code. It is loaded automatically if the PDF management code is active. The name of the driver will change after the testphase.

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

The driver *requires* the new PDF management code, so documents wanting to use it should start like this (this requires L^AT_EX-2022-06-01 or newer):

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

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

1 Avoiding transition problems

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

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

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

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

2 Bookmarks / outlines

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

3 “Metadata”

“Metadata”, information about the document, are stored in a PDF in two places: The `/Info` dictionary and the XMP-metadata. `hyperref` only handles the `/Info` dictionary. The XMP-metadata are added by code from `l3pdfmeta`. (without the `pdfmanagement` the XMP-metadata can be added with packages like `pdfx` and `hyperxmp`).

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

The current handling of the metadata is problematic:

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

To resolve some of this problem the driver will

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

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

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

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

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

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

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

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

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

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

4 Dates

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

5 PDF page size (mediabox)

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

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

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

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

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

6 Commands to create “external” references

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

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

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

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

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

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

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

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

6.1 Special problem: links to files

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

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

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

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

6.2 Splits

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

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

For this reason three new commands have been create:

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

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

6.3 Options

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

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

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

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

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

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

Some main changes are

- The default colors have been changed.

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

7.1 Background information

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

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

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

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

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

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

7.2 New Keys

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

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

- Beside `pdfhighlight` there are also `linkhighlight`, `urlhighlight` etc
- Beside `pdfborderstyle` there are also `linkborderstyle`, `urlborderstyle` etc
- Beside `colorlinks` there are also `colorlink`, `colorurl` etc
- Beside `ocgcolorlinks` there are also `ocgcolorlink`, `ocgcolorurl`, etc TODO
- Beside `hidelinks` there are also `hidelink`, `hideurl`, etc
- `bordercolormodel` allows to set the model used in annotations, the allowed values are `rgb` or `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), `phetype`, `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 `phetype`.

7.3 Public interfaces

```

\l_hyp_annot_colorlink_bool
\l_hyp_annot_colorurl_bool
\l_hyp_annot_colorfile_bool
\l_hyp_annot_colorrurl_bool
\l_hyp_annot_colormenu_bool
\l_hyp_annot_ocgcolorlink_bool
\l_hyp_annot_ocgcolorurl_bool
\l_hyp_annot_ocgcolorfile_bool
\l_hyp_annot_ocgcolorrurl_bool
\l_hyp_annot_ocgcolormenu_bool

```

These boolean are used by the `colorlinks` and `ocgcolorlinks` and related keys. These keys insert hook code in the `pdfannot/link/<type>/begin` and `pdfannot/link/<type>/end` hooks. `<type>` is one of `GoTo`, `URI`, `GoToR`, `Named` or `Launch`.

`colorlinks` uses the label `hyp/color`, and `ocgcolorlinks` the label `hyp/ocg`.

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

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

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

7.4 Changed behaviour

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

Color keys accept the following input syntax:

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from hyperref

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

Options that still must be set as package options are

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

Options that can be without problems set as package options are

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

Ignored options:

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

10 Disabling links

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

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

`nested-links` (*setup key*)

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

11 Draftmode

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

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

12 Dropped options

A number of options are ignored by this driver

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

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

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

unicode This is always true.

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

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

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

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

13 Destinations

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

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

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

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

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

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

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

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

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

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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

`nesting` (*setup key*)

`pdfborder` (*setup key*) This key set accept as value three numbers or three numbers and an array describing

`linkborder` (*setup key*) a dash pattern, examples are 0 0 1 or 0 0 1 [3 2]. The first two numbers should

`urlborder` (*setup key*) according to the reference set round corners, but PDF viewer seem to ignore it. The

`runborder` (*setup key*) third number is the line width of the border. Settings done with `pdfborderstyle` should

`menuborder` (*setup key*) take precedence.

`pdfborderstyle` (*setup key*) The value of this key is the content of the BS dictionary. As an example

`linkborderstyle` (*setup key*) `/Type/Border /W 1 /S/U /D[3 2]`

	Key	Values	comment / example
<code>urlborderstyle</code> (<i>setup key</i>)	<code>/Type</code>	<code>/Border</code>	optional
<code>fileborderstyle</code> (<i>setup key</i>)	<code>/W</code>	<code><number></code>	Width of border line
<code>runborderstyle</code> (<i>setup key</i>)	<code>/S</code>	<code>/S</code>	solid (default)
<code>menuborderstyle</code> (<i>setup key</i>)		<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])

`pdfcreationdate` (*setup key*) Setting these keys is normally not needed. If they are used the values of the first

`pdfmoddate` (*setup key*) two keys are stored directly in the Info dictionary for `/Creationdate` and `/ModDate`.

`pdfmetadate` (*setup key*) All three keys are used in XMP-metadata. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. `D:20200202111111+01'00'`.

`pdflinkmargin` (*setup key*) As described in the `hyperref` manual the behaviour differs between the backends: with dvips it is possible to change links locally, pdf_{flat} and lua_{latex} work by page, with dvipdfmx the setting is global (and has to be done in the preamble).

`pdflang` (*setup key*) The key will work, but it is recommended to set the language in `\DocumentMetadata` instead.

File I

hyperref-generic driver implementation

```

1 <@@=hyp>
2 <*headertestphase>
3 \ProvidesFile{hggeneric-testphase.def}[2025-06-23 v0.96s %
4   generic Hyperref driver for the LaTeX PDF management bundle]
5 </headertestphase>
6 <*header>
7 \ProvidesFile{hggeneric.def}[2025-06-23 v0.96s %
8   generic Hyperref driver for the LaTeX PDF management bundle]
9 </header>
10 <*package>
11 \RequirePackage{etoolbox} %why?

```

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

```

12 \long\def\Hy@ReturnAfterFi#1\fi{\fi#1}
13 \ExplSyntaxOn
14 \file_input:n {hyperref-colorschemes.def}
15 \ExplSyntaxOff

```

1 messages

Redirect the message name:

```

16 \ExplSyntaxOn
17 \prop_gput:Nnn \g_msg_module_name_prop { hyp }{ hyperref }

```

At first a message for the testing of the resource management

```

18 \cs_if_exist:NTF \DocumentMetadata
19 {
20   \msg_new:nnnn
21     { hyp }
22     { missing-resource-management }
23     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
24     {
25       Activate~it~with ~\
26       \tl_to_str:n{\DocumentMetadata{<options>}}\
27       before~\tl_to_str:n{\documentclass}
28     }
29 }
30 {
31   \msg_new:nnnn
32     { hyp }
33     { missing-resource-management }
34     { The~PDF~resource~management~is~required~for~this~hyperref~driver! }
35     {
36       Activate~it~with ~\
37       \tl_to_str:n{\DocumentMetadata{<options>}}\
38       before~\tl_to_str:n{\documentclass} or ~\
39       \RequirePackage{pdfmanagement}
40     }
41 }

```

The pdfversion should be set in \DocumentMetadata

```

42 \msg_new:nnnn
43   { hyp }
44   { pdfversion-disabled }
45   {
46     This~hyperref~driver~ignores~the~pdfversion~key!\
47     Set~the~pdfversion~in~\token_to_str:N \DocumentMetadata
48   }
49   {
50     For example:\
51     \tl_to_str:n
52     {
53       \DocumentMetadata { pdfversion=1.7 }
54     }
55   }

```

A generic message for ignored keys.

```
56 \msg_new:nnn
57 { hyp }
58 { key-dropped }
59 {
60   This~hyperref~driver~ignores~the~key~#1!\\
61   Please~check~the~documentation.
62 }
```

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

```
63 \msg_new:nnn
64 { hyp }
65 { pdfa-no-push-button }
66 { PDF/A:~Push~button~with~JavaScript~is~prohibited }
67
68 \msg_new:nnn
69 { hyp }
70 { pdfa-no-reset-button }
71 { PDF/A:~Reset~action~is~prohibited }
```

pdf/A message for not allowed Named actions

```
72 \msg_new:nnn
73 { hyp }
74 { pdfa-no-named-action }
75 { PDF/A:~Named~action~#1~is~prohibited }
```

A message if the destination name is empty.

```
76 \msg_new:nnn
77 { hyp }
78 { empty-destination-name }
79 {
80   Empty~destination~name,\\
81   using~'#1'
82 }
```

A message if the destination check fails

```
83 \msg_new:nnn
84 { hyp }
85 { invalid-destination-value }
86 {
87   Invalid~value~'#1'~of~'#2'  \\
88   is~replaced~by~'Fit'~\msg_line_context:.
89 }
```

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

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



```

101     Value~'#1'~is~unknown~or~deprecated~in\\
102     pdf~version~#2.~Ignored.
103 }
104 \msg_new:nnn
105 { hyp }
106 { replace-deprecated-or-unknown-value-in-pdf-version }
107 {
108     Value~'#1'~is~unknown~or~deprecated~in\\
109     pdf~version~#2. Value~'#3'~is used instead.
110 }

```

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:

```

111 \msg_new:nnn
112 { hyp }
113 { unknown-key }
114 {
115     unknown-key~#2~of~module~'#1'~set~to~'#3'.
116 }
117 \msg_new:nnn
118 { hyp }
119 { unknown-key-to-Hyp }
120 {
121     ignored-in-family-Hyp-unknown-key~#1.
122 }

```

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

```

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

```

```

150     *~false \\
151     *~and~an~empty~value~which~removes~the~setting.\\
152     No~value~is~equivalent~to~using~'true'.
153 }

```

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

```

154 \msg_new:nnn
155 { hyp }
156 { empty-info-value }
157 {
158     Empty~value~for~key~#1.\\
159     This~isn't~honored~by~all~backends.
160 }

```

2 Variants

```

161 \cs_generate_variant:Nn\pdf_destination:nn {nf}
162 \cs_generate_variant:Nn\pdf_object_ref:n {e}
163 \cs_generate_variant:Nn\pdf_pageobject_ref:n {e}

```

3 Overwriting/providing commands from hyperref

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

```

164 \chardef\Hy@VersionChecked=1 %don't check the version!
165 %\cs_set_protected:Npn \PDF@SetupDoc{}
166 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

\hypercalcbp

```

167 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

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

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

```

168 \providecommand\@pdfborder{}
169 \providecommand\@pdfborderstyle{}
170 \newcommand\OBJ@OCG@view {} % needed in hyperref
171 \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:

```

172 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
173 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
174 \legacy_if:nT { Hy@setpdfversion }
175 {
176     \msg_warning:nn { hyp }{ pdfversion-disabled }
177 }
178 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

179 \RenewDocumentCommand \Acrobatmenu { m m }
180 {
181   \hyper@linknamed {#1} {#2}
182 }

```

\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!

```

183 % TODO should go at some time ...
184 % \kv@set@family@handler{Hyp}
185 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
186 \cs_set_protected:Npn \hypersetup #1
187 {
188   %\kvsetkeys{Hyp} {#1}
189   \keys_set:nn { hyp }{ #1 }
190 }
191 % TODO for now unknown keys should only give warnings.
192 \keys_define:nn { hyp }
193 {
194   unknown .code:n =
195   {
196     \msg_warning:nneee { hyp } { unknown-key }
197     { hyp }{ \l_keys_key_str } { #1 }
198   }
199 }

```

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.

```

200 \keys_define:nn { hyp }
201 {
202   next-anchor .code:n =
203   {
204     \AddToHookNext{__hyp/dest/make}
205     {\Hy@MakeCurrentHref{#1}}
206   }
207 }

```

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`)

```

208
209 \bool_new:N \l__hyp_href_url_encode_bool
210 \bool_new:N \l__hyp_href_url_ismap_bool
211 \tl_new:N \l__hyp_href_url_protocol_tl
212 \tl_new:N \l__hyp_href_pdf_destination_tl
213 \tl_new:N \l__hyp_href_pdf_page_tl
214 \tl_new:N \l__hyp_href_run_parameter_tl
215 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\url}
216
217
218 \keys_define:nn { hyp / href }

```

```

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

```

```

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

```

```

327
328 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
329 {
330   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
331   \group_end:
332   \hook_use:n{cmd/hrefpdf/after}
333 }
334
335 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
336 \DeclareRobustCommand*{\hrefrun}[1] []
337 {
338   \mode_leave_vertical:
339   \hook_use:n{cmd/hrefrun/before}
340   \group_begin:
341   \keys_set:nn { hyp / href } {#1}
342   \hyper@normalise\__hyp_href_run_aux:nn
343 }
344
345 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
346 {
347   \exp_args:Nnno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
348   \group_end:
349   \hook_use:n{cmd/hrefrun/after}
350 }
351
352
353 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}
354
355 \DeclareRobustCommand*{\url}[1] []
356 {
357   \mode_leave_vertical:
358   \hook_use:n{cmd/url/before}
359   \group_begin:
360   \keys_set:nn { hyp / href } {#1}
361   \bool_if:NTF \l__hyp_href_url_encode_bool
362   {
363     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
364   }
365   {
366     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
367   }
368   \hyper@normalise\__hyp_href_url_aux:n
369 }
370
371 \cs_new_protected:Npn \__hyp_href_url_aux:n #1
372 {
373   \exp_args:Nno
374   \hyper@linkurl{\__hyp_href_url_format: {#1}}
375   {\l__hyp_href_url_protocol_tl#1}
376   \group_end:
377   \hook_use:n{cmd/url/after}
378 }
379

```

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

`\url` command here:

```

380
381 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
382 \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
383 % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
384 \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
385 #2}
386

```

make the new commands compatible with `\pdfstringdef`:

```

387 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
388 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
389 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
390 \pdfstringdefDisableCommands{\let\hrefrun\__hyp_secondoftwowithopt:wnn}

```

4 Compatibility commands

4.1 Metadata

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

```

391 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
392 {
393   %\tl_set:cn {@#1}{#2}
394   \AddToDocumentProperties[hyperref]{#1}{#2}
395 }
396 \cs_generate_variant:Nn \__hyp_store_metadata:nn {en,ne,ee,no,eo}

```

4.2 citecolor

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

```

397 \hook_new:n{hyp/link/cite}
398 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
399 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
400 \keys_define:nn { hyp }
401 {
402   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init
403   ,citebordercolor
404   .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_i
405 }
406 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
407 {
408   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
409   {
410     \keys_set:nn { hyp }
411     {
412       linkcolor      = hyp/color/cite
413     }
414   }
415   \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
416 }

```

```

417 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
418 {
419   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
420   {
421     \keys_set:nn { hyp }
422     {
423       linkbordercolor      = hyp/color/citeborder
424     }
425   }
426   \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
427 }
428

```

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

```

429 \bool_lazy_and:nnF
430 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
431 { \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

```

432 \AddToHook { package/bookmark/after}
433 {
434   \define@key{BKM}{color}
435   {
436     \tl_if_blank:nTF {#1}
437     { \cs_set_eq:NN\BKM@color\@empty }
438     {
439       \__hyp_color_set:ne {__hyp/tmpa}{#1}
440       \color_export:nVN
441       {__hyp/tmpa}
442       \g__hyp_bordercolormodel_str
443       \BKM@color
444     }
445   }
446 }
447 \legacy_if:nTF { Hy@bookmarks }
448 {
449   \AddToHook{begindocument/before}[hyperref/bookmark]
450   {
451     \RequirePackage{bookmark}
452   }
453 }

```

empty hook chunk to ensure that the chunk exists.

```

454 {
455   \AddToHook{begindocument/before}[hyperref/bookmark]{}
456 }

```



```

457 \legacy_if:nT {Hy@draft}
458 {
459   \PassOptionsToPackage{draft}{bookmark}
460 }

```

6 Reference and label commands

This uses the in-built property module.

```
\__hyp_property_record:nn
```

```

461 %
A label command which adds the space commands from LaTeX:
462 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
463 {
464   \@bsphack
465   \property_record:nn{#1}{#2}
466   \@esphack
467 }

```

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

```
468 \cs_generate_variant:Nn \__hyp_property_record:nn {ee}
```

(End of definition for __hyp_property_record:nn.)

7 Variables

7.1 Private temporary variables

At first a few generic tmp variables

```

\l__hyp_tmpa_tl
\l__hyp_tmpb_tl
\l__hyp_tmpa_seq
\l__hyp_tmpa_int
\l__hyp_tmpa_box
\l__hyp_tmpa_str
469 \box_new:N \l__hyp_tmpa_box
470 \tl_new:N \l__hyp_tmpa_tl
471 \tl_new:N \l__hyp_tmpb_tl
472 \seq_new:N \l__hyp_tmpa_seq
473 \int_new:N \l__hyp_tmpa_int
474 \str_new:N \l__hyp_tmpa_str

```

(End of definition for \l__hyp_tmpa_tl and others.)

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

TODO: document and check use!

```

\l__hyp_dest_name_tmpa_tl
\l__hyp_uri_tmpa_tl
\l__hyp_filename_tmpa_tl
__hyp_text_tmpa_str \g__hyp_text_tmpa_str
475 \tl_new:N \l__hyp_dest_name_tmpa_tl
476 \tl_new:N \l__hyp_uri_tmpa_tl
477 \tl_new:N \l__hyp_filename_tmpa_tl
478 \tl_new:N \l__hyp_para_tmpa_tl
479 \str_new:N \l__hyp_text_tmpa_str
480 \str_new:N \g__hyp_text_tmpa_str

```

(End of definition for \l__hyp_dest_name_tmpa_tl and others.)

7.2 Constants

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

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

(End of definition for \c__hyp_dest_undefined_tl.)

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

```
482 \seq_const_from_clist:Nn \c__hyp_annot_types_seq
483 {url,link,file,menu,run}
484 \prop_const_from_keyval:Nn \c__hyp_map_annot_hyp_prop
485 {
486   URI    = url,
487   GoTo   = link,
488   GoToR  = file,
489   Named  = menu,
490   Launch = run
491 }
492 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
493 {
494   url    = URI,
495   link   = GoTo,
496   file   = GoToR,
497   menu   = Named,
498   run    = Launch
499 }
500
```

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

7.3 Variables

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

```
501 \tl_new:N \g__hyp_dest_pdfstartpage_tl
502 \tl_new:N \g__hyp_dest_pdfstartview_tl
503 \tl_new:N \l__hyp_dest_pdfremotestartview_tl
```

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

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

```
\l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_info_print_tl
504 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
505 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
506 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
507 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
508 \tl_new:N \l__hyp_text_enc_file_print_tl
509 \tl_new:N \l__hyp_text_enc_para_print_tl
```

```

510
511 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
512 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
513 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
514 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
515 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
516 \tl_set:Nn \l__hyp_text_enc_para_print_tl {utf8/string}

```

(End of definition for \l__hyp_text_enc_uri_print_tl and others.)

It is also unclear how the /Contents entry would look at best. So we use sockets. The first argument is the target (url or destination), For GoTo we also pass the text as argument. The sockets should put something into the relevant annotation dictionaries.

```

517 \tl_new:N\l__hyp_link_Contents_tl
518 \socket_new:nn {hyp/link/GoTo/Contents}{2}
519 \socket_new:nn {hyp/link/URI/Contents}{1}
520 \socket_new:nn {hyp/link/GoToR/Contents}{1}
521 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}
522 {
523   \__hyp_text_pdfstring:eoN
524   { Go~to~destination~#1 }
525   { \l__hyp_text_enc_info_print_tl }
526   \l__hyp_link_Contents_tl
527   \pdfannot_dict_put:nne {link/GoTo}{Contents}
528   {\l__hyp_link_Contents_tl}
529 }
530 \socket_new_plug:nnn {hyp/link/GoToR/Contents}{default}
531 {
532   \__hyp_text_pdfstring:eoN
533   { Open~file~#1 }
534   { \l__hyp_text_enc_info_print_tl }
535   \l__hyp_link_Contents_tl
536   \pdfannot_dict_put:nne {link/GoToR}{Contents}
537   {\l__hyp_link_Contents_tl}
538 }
539 \socket_new_plug:nnn {hyp/link/URI/Contents}{default}
540 {
541   \__hyp_text_pdfstring:eoN
542   { #1 }
543   { \l__hyp_text_enc_info_print_tl }
544   \l__hyp_link_Contents_tl
545   \pdfannot_dict_put:nne {link/URI}{Contents}
546   {\l__hyp_link_Contents_tl}
547 }
548 \socket_assign_plug:nn{hyp/link/GoTo/Contents}{default}
549 \socket_assign_plug:nn{hyp/link/GoToR/Contents}{default}
550 \socket_assign_plug:nn{hyp/link/URI/Contents}{default}

```

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

```

551 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End of definition for \l__hyp_dest_pdfview_tl.)

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

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

```
552 \str_new:N \g__hyp_bordercolormodel_str
```

(End of definition for \g__hyp_bordercolormodel_str.)

7.4 Booleans

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

```
553 \seq_map_inline:Nn \c__hyp_annot_types_seq
554 {
555   \bool_new:c {l_hyp_annot_color#1_bool}
556 }
```

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

`\l_hyp_annot_ocgcolorlink_bool` `\l_hyp_annot_ocgcolorurl_bool` `\l_hyp_annot_ocgcolorfile_bool` `\l_hyp_annot_ocgcolorrerun_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.

```
557 \seq_map_inline:Nn \c__hyp_annot_types_seq
558 {
559   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
560 }
```

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

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

```
561 \seq_map_inline:Nn \c_pdfannot_link_types_seq
562 {
563   \bool_new:c {l__hyp_annot_#1_bool}
564   \bool_set_true:c {l__hyp_annot_#1_bool}
565 }
```

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

7.5 Boxes

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

```
566 \box_new:N \l__hyp_dest_box
```

(End of definition for \l__hyp_dest_box.)

7.6 Regex

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

```

567 \regex_const:Nn \c__hyp_dest_startview_regex
568 {
569   \A\ *
570   (?:
571     (?:XYZ (?:\ +(?:\d+|\d*\.\d+)|null)){3}\ )
572     |
573     (?:Fit\b|FitB\b)
574     |
575     (?:\b(?:FitH|FitV|FitBH|FitBV)(?:\ +(?:\d+|\d*\.\d+)|\ +null){1})
576     |
577     (?:FitR (?:\ +\d+|\ +\d*\.\d+){4}\ )
578   )
579 }
```

(End of definition for \c__hyp_dest_startview_regex.)

7.7 PDF dictionaries

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

```

580 \pdfdict_new:n {l__hyp_page/Trans}
581 \pdfdict_put:nnn {l__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for l__hyp_page/Trans.)

8 PDF string conversion

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

`hyp/text/pdfstring`

```

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

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

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

`__hyp_text_purify:nN`

```

583 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
584 {
585   \str_set:Ne #2 {\text_purify:n { #1 } }
586 }
```

(End of definition for __hyp_text_purify:nN.)

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

`__hyp_text_cleanup:N`

```
587 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
588 {
589
590 }
```

(End of definition for __hyp_text_cleanup:N.)

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

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

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

`__hyp_text_string_from_unicode:nN`

```
591 \cs_new_protected:Npn \__hyp_text_string_from_unicode:nN #1 #2
592 {
593   \pdf_string_from_unicode:nVN { #1 } #2 #2
594 }
```

(End of definition for __hyp_text_string_from_unicode:nN.)

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

`__hyp_text_pdfstring:nnN`

```
595 \cs_new_protected:Npn \__hyp_text_pdfstring:nnN #1 #2 #3
596 {
597   \group_begin:
598   \Hy@pdfstringtrue
599   \hook_use:n {hyp/text/pdfstring}
600   \__hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
601   \__hyp_text_cleanup:N      \l__hyp_text_tmpa_str
602   \__hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
603   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
604   \group_end:
605   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
606 }
607 \cs_generate_variant:Nn \__hyp_text_pdfstring:nnN {enN,onN,eoN,ooN,noN}
```

(End of definition for __hyp_text_pdfstring:nnN.)

!!! temporary until all instances are gone

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

This is a special version for info keys:

```

\__hyp_text_pdfstring_info:nN
610 \cs_new_protected:Npn \__hyp_text_pdfstring_info:nN #1 #2
611 {
612   \__hyp_text_pdfstring:non { #1 }{ \l__hyp_text_enc_info_print_tl } #2
613 }
614 \cs_generate_variant:Nn \__hyp_text_pdfstring_info:nN {eN,oN}
(End of definition for \__hyp_text_pdfstring_info:nN.)

```

9 Pagelabels

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

The current implementation in hyperref/hyperref drivers:

xetex: hxdetex.def, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in hyperref inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-31 is correctly split. If the counter can not be identified hyperref generates only `/P` entries with the whole content.

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

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

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

```

\__hyp_PageLabels_gpush:
  \Hy@PutCatalog
  \HyPL@StorePageLabel
615 \cs_new_protected:Npn \__hyp_PageLabels_gpush:
616 {
617   \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums[\HyPL@Labels]>>}
618 }
619
620 \def\Hy@PutCatalog #1 {}
621
622
623 \legacy_if:nT { Hy@pdfpagelabels }

```

```

624 {
625   \cs_set_protected:Npn \HyPL@StorePageLabel #1
626   {
627     \tl_gput_right:Nx \HyPL@Labels { \the\Hy@abspage<<#1>> }
628     \__hyp_PageLabels_gpush:
629   }
630 }

```

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

10 Core Hyperref Commands

Every hyperref has to define eight core command:

```

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

```

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

10.1 Link level

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

`g__hyp_linknestlevel_int`

```

631 \int_new:N \g__hyp_linknestlevel_int

```

(End of definition for `g__hyp_linknestlevel_int`.)

```

632 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
633 {
634   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
635   { \prg_return_false: }
636   { \prg_return_true: }
637 }
638 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
639 {
640   \use_i:nn {#1}{#2}
641 }
642 \keys_define:nn { hyp }
643 {
644   nested-links .choice:,
645   nested-links / true .code:n =
646   { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },

```



```

647     nested-links / false .code:n =
648     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
649     nested-links .default:n = {true}
650 }

```

10.2 Anchors / destinations

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

```

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

```

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

```

\__hyp_destination:nn
651 \Hy@WrapperDef \__hyp_destination:nn #1 #2
652 {
653   \mode_if_horizontal:T { \@savs\spacefactor }
654   \Hy@SaveLastskip      %defined in hyperref
655   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging
656   \__hyp_text_pdfstring:eoN
657   { \HyperDestNameFilter{#1} }
658   { \l__hyp_text_enc_dest_tl }
659   \l__hyp_tmpa_tl
660   \str_if_eq:nnTF {#2} {fitrbx}
661   {
662     \exp_args:NV
663     \pdf_destination:nnnn \l__hyp_tmpa_tl
664     { \box_wd:N \l__hyp_dest_box }
665     { \box_ht:N \l__hyp_dest_box }
666     { \box_dp:N \l__hyp_dest_box }
667   }
668   {
669     \exp_args:NV
670     \pdf_destination:nf
671     { \l__hyp_tmpa_tl }
672     { #2 }
673   }
674   \Hy@RestoreLastskip %defined in hyperref
675   \mode_if_horizontal:T { \spacefactor\@savs }
676 }

```

(End of definition for __hyp_destination:nn.)

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

```

\hyper@anchor
\hyper@anchorstart 677 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 678 \hook_new:n{hyp/anchor}
hyp/anchor 679 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 680 {
681   \exp_args:NnV
682   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
683   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
684   \hook_use:n{hyp/anchor}
685 }
686
687 \cs_new_protected:Npn \hyper@anchorstart #1
688 {
689   \Hy@activeanchortrue
690   \exp_args:NnV
691   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
692   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
693   \hook_use:n{hyp/anchor}
694 }
695
696 \cs_new_protected:Npn \hyper@anchorend
697 {
698   \Hy@activeanchorfalse
699 }

```

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

10.3 GoTo Links

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

At first the internal link commands:

```

700 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
701 {
702   \mode_leave_vertical:
703   \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
704   \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
705   {
706     \msg_warning:nne
707     { hyp }
708     { empty-destination-name }
709     { \c__hyp_dest_undefined_tl }
710     \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
711   }
712   {
713     \__hyp_text_pdfstring:eoN
714     { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
715     { \l__hyp_text_enc_dest_tl }
716     \l__hyp_dest_name_tmpa_tl
717   }
718   \exp_args:No

```

```

719     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
720 }
721
722 \cs_new_protected:Npn \__hyp_link_goto_end:
723 {
724     \pdfannot_link_goto_end:
725 }

```

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

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

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

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

```

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

```

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

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

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

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

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

```

726
727 \cs_new_protected:Npn \hyper@link #1 #2 #3 % #1 context, #2=destination name, #3 content
728 {
729     \bool_if:NTF \l__hyp_annot_GoTo_bool
730     {
731         \int_gincr:N \g__hyp_linknestlevel_int
732         \__hyp_check_link_nesting:TF
733         {
734             \Hy@VerboseLinkStart{#1}{#2}
735             \group_begin:
736             \tl_set:Nn \l_hyp_current_dest_name_tl {#2}

```

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

```

737         \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{#3}
738         \hook_use:n {hyp/link/#1}
739         \__hyp_link_goto_begin:nw {#2} #3 \Hy@xspace@end
740         \__hyp_link_goto_end:
741         \group_end:
742         \Hy@VerboseLinkStop
743     }
744     {
745         \group_begin: #3 \group_end:

```

```

746     }
747     \int_gdecr:N\g__hyp_linknestlevel_int
748   }
749   {{\let\protect\relax#3}}
750 }
751 \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
752 {
753   \bool_if:NT \l__hyp_annot_GoTo_bool
754   {
755     \int_gincr:N\g__hyp_linknestlevel_int
756     \__hyp_check_link_nesting:TF
757     {
758       \Hy@VerboseLinkStart{#1}{#2}% only for debug
759       \group_begin:
760       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
761       \socket_use:nnn{hyp/link/GoTo/Contents}{#2}{ }
762       \hook_use:n {hyp/link/#1}
763       \__hyp_link_goto_begin:nw {#2}
764     }
765     {
766       \group_begin:
767     }
768   }
769 }
770
771 \cs_new_protected:Npn \hyper@linkend
772 {
773   \bool_if:NT \l__hyp_annot_GoTo_bool
774   {
775     \__hyp_check_link_nesting:TF
776     {
777       \__hyp_link_goto_end:
778       \group_end:
779       \Hy@VerboseLinkStop
780     }
781     {
782       \group_end:
783     }
784     \int_gdecr:N\g__hyp_linknestlevel_int
785   }
786 }

```

10.4 URI links

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

```

787 \pdfdict_new:n {l_hyp/annot/A/URI}
788 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
789 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
790
791 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
792 {
793   \bool_if:NTF \l__hyp_annot_URI_bool
794   {
795     \int_gincr:N\g__hyp_linknestlevel_int

```

```

796     \__hyp_check_link_nesting:TF
797     {
798         \group_begin:
799         \__hyp_text_pdfstring:eoN
800         { #2}
801         { \l__hyp_text_enc_uri_print_tl }
802         \l__hyp_uri_tmpa_tl
803         \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
804         \bool_if:NT \l__hyp_href_url_ismap_bool
805         {
806             \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
807         }

```

This socket adds something to the /Contents key.

```

808     \socket_use:nn{hyp/link/URI/Contents}{#2}
809     \cs_set_eq:NN \# \c_hash_str
810     \cs_set_eq:NN \% \c_percent_str
811     \Hy@safe@activestrue
812     \mode_leave_vertical:
813     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
814     \pdfannot_link:nen { URI }
815     {
816     }
817     {
818         \let\protect\relax
819         #1
820         \Hy@xspace@end
821         \Hy@VerboseLinkStop %where is the start??
822     }
823     \group_end:
824     }
825     {
826         \group_begin: #1 \group_end:
827     }
828     \int_gdecr:N\g__hyp_linknestlevel_int
829 }
830 {{\let\protect\relax#1}}
831 }
832

```

10.5 GoToR Links files

```

833 \pdfdict_new:n {l_hyp/annot/A/GoToR}
834 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
835 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
836
837 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
838 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
839 {
840     \bool_if:NTF \l__hyp_annot_GoToR_bool
841     {
842         \int_gincr:N\g__hyp_linknestlevel_int
843         \__hyp_check_link_nesting:TF
844         {

```

```

845 \group_begin:
846 \tl_set:Nc \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
847 \exp_args:Nc
848 \pdf_object_if_exist:nF { __hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl }
849 {
850 \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n
851 \pdffile_embed_file:noe
852 {}
853 {\l__hyp_filename_tmpa_tl }
854 {__hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl }
855 }
856 \pdfdict_put:nne
857 {l_hyp/annot/A/GoToR}
858 {F}
859 {\pdf_object_ref:e {__hyp_file\__tl_to_str:N \l__hyp_filename_tmpa_tl}}
860 \__hyp_text_pdfstring:nnN
861 { #3 }
862 { \l__hyp_text_enc_dest_print_tl }
863 \l__hyp_dest_name_tmpa_tl

```

This socket adds something to the /Contents key.

```

864 \socket_use:nn{hyp/link/GoToR/Contents}{#2}
865 \tl_if_blank:eTF {#3}
866 {
867 \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
868 {
869 [
870 \int_eval:n
871 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
872 /\l__hyp_dest_pdfremotestartview_tl
873 ]
874 }
875 }
876 {
877 \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
878 }
879 \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>

```

880 \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
881 \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
882 \pdfannot_link:nnn %expansion??
883 { GoToR }
884 {
885 }
886 {
887 \let\protect\relax
888 #1\Hy@xspace@end
889 \Hy@VerboseLinkStop %where is the start??
890 }
891 \group_end:
892 }
893 {
894 \group_begin: #1 \group_end:

```

```

895     }
896     \int_gdecr:N\g__hyp_linknestlevel_int
897   }
898   {{\let\protect\relax#1}}
899 }

```

10.6 Launch links

We define \hyper@linklaunch for naming consistency

```

900 \pdfdict_new:n {l_hyp/annot/A/Launch}
901 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
902 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
903
904 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
905 {
906   \bool_if:NTF \l__hyp_annot_Launch_bool
907   {
908     \int_gincr:N\g__hyp_linknestlevel_int
909     \__hyp_check_link_nesting:TF
910     {
911       \group_begin:
912       \__hyp_text_pdfstring:nnN
913       { #1 }
914       { \l__hyp_text_enc_file_print_tl }
915       \l__hyp_filename_tmpa_tl
916       \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
917       \__hyp_text_pdfstring:noN
918       { #3 }
919       { \l__hyp_text_enc_para_print_tl }
920       \l__hyp_para_tmpa_tl
921       \bool_if:NTF
922       {
923         \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}{}
924         ||
925         \pdf_version_compare_p:Nn > {1.9}
926       }
927       {
928         \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
929       }
930       {
931         \pdfdict_put:nne
932         {l_hyp/annot/A/Launch}
933         {Win}
934         {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
935       }
936       \mode_leave_vertical:
937       \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
938       \pdfannot_link:nen
939       { Launch }
940       {
941         % /A
942         % <<
943         % \pdfdict_use:n {l_hyp/annot/A/Launch}
944         % >>
945       }

```

```

946         {
947             \let\protect\relax
948             #2\Hy@xspace@end
949             \Hy@VerboseLinkStop %where is the start??
950         }
951     \group_end:
952 }
953 { \group_begin: #2 \group_end: }
954 \int_gdecr:N\g__hyp_linknestlevel_int
955 }
956 {\let\protect\relax#2}}
957 }

```

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

```

958 \use:e
959 { % filename, anchor text, linkname
960   \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
961 }
962 {
963   \hyper@linklaunch {#1}{#2}{#3}
964 }

```

10.7 Named links (menu)

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

```

965 \pdfdict_new:n {l_hyp/annot/A/Named}
966 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
967 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
968
969 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
970 {
971   \bool_if:NTF \l__hyp_annot_Named_bool
972   {
973     \int_gincr:N\g__hyp_linknestlevel_int
974     \__hyp_check_link_nesting:TF
975     {
976       \group_begin:
977       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
978       {
979         \mode_leave_vertical:
980         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
981           {\pdf_name_from_unicode_e:n{#1}}
982         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
983         \pdfannot_link:nnn { Named }
984         {
985           % /A
986           % <<
987           % \pdfdict_use:n { l_hyp/annot/A/Named }
988           % >>
989         }
990       }
991       #2
992       \Hy@xspace@end

```



```

993             \Hy@VerboseLinkStop
994         }
995     }
996     {
997         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
998         #2
999     }
1000     \group_end:
1001 }
1002 { \group_begin: #2 \group_end: }
1003 \int_gdecr:N\g__hyp_linknestlevel_int
1004 }
1005 {{\let\protect\relax#2}}
1006 }
1007

```

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.

```

1008 \cs_generate_variant:Nn \color_export:nnN {nVN}

```

```

\__hyp_color_select:n \__hyp_color_select:n {<color>}

```

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

```

\__hyp_color_select:n \__hyp_color_select:n {<color>}
\__hyp_color_select_aux:wn

```

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

```

1009 \cs_new_protected:Npn \__hyp_color_select:n #1
1010 {
1011     \tl_if_head_eq_charcode:nNTF {#1}[ %]
1012     {
1013         \__hyp_color_select_aux:wn #1
1014     }
1015     {
1016         \color_select:n {#1}
1017     }
1018 }
1019
1020 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
1021 {
1022     \color_select:nn {#1}{#2}
1023 }
1024
1025 \cs_generate_variant:Nn \__hyp_color_select:n {e}

```

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

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

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

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

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

```
1026 \cs_new_protected:Npn \_hyp_color_set:nn #1 #2
1027 {
1028   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1029   {
1030     \_hyp_color_set_aux:nwn { #1 } #2
1031   }
1032   {
1033     \color_set:nn {#1} {#2}
1034   }
1035 }
1036
1037 \cs_new_protected:Npn \_hyp_color_set_aux:nwn #1 [#2] #3
1038 {
1039   \color_set:nnn {#1}{#2}{#3}
1040 }
1041
1042 \cs_generate_variant:Nn \_hyp_color_set:nn {ne}
```

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

11.2 Textcolor of links

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

```
1043 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1044 {
1045   \hook_gput_code:nnn
1046     {pdfannot/link/#2/begin}
1047     {hyp/color}
1048     {
1049       \bool_if:cT { l_hyp_annot_color#1_bool }
1050       {
1051         \group_begin:
1052         \color_select:n { hyp/color/#1}
1053       }
1054     }
1055   \hook_gput_code:nnn
1056     {pdfannot/link/#2/end}
1057     {hyp/color}
1058     {
1059       \bool_if:cT { l_hyp_annot_color#1_bool }
1060       {
1061         \group_end:
1062       }
1063     }
1064 }
```

```

1063     }
1064 }

```

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

```

1065 \keys_define:nn { hyp }
1066 {
1067   ,colorlinks .choice:
1068   ,colorlinks / true .meta:n =
1069   {
1070     ,pdfborder={0~0~0}
1071     ,pdfborderstyle=
1072     ,colorurl =#1
1073     ,colorlink =#1
1074     ,colorrurl =#1
1075     ,colormenu =#1
1076     ,colorfile =#1
1077   }
1078   ,colorlinks / false .meta:n =
1079   {
1080     ,colorurl =#1
1081     ,colorlink =#1
1082     ,colorrurl =#1
1083     ,colormenu =#1
1084     ,colorfile =#1
1085   }
1086   ,colorlinks .default:n = {true}
1087 }

```

`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)`

```

1088 \seq_map_inline:Nn \c__hyp_annot_types_seq
1089 {
1090   \keys_define:nn { hyp }
1091   {
1092     ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
1093     ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
1094   }
1095 }
1096
1097 \keys_define:nn { hyp }
1098 {
1099   ,allcolors .meta:n =
1100   {
1101     ,urlcolor=#1
1102     ,linkcolor=#1
1103     ,runcolor=#1
1104     ,filecolor=#1
1105     ,menucolor=#1
1106   }
1107   ,allcolors .value_required:n = true
1108 }

```

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*)

```
1109 \keys_define:nn { hyp }
1110 {
1111   ,bordercolormodel .choices:nn =
1112   {rgb,cmyk}
1113   { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1114   ,bordercolormodel .initial:n = {rgb}
1115 }

1116 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1117 {
1118   \keys_define:nn { hyp }
1119   {
1120     #1bordercolor .code:n =
1121     {
1122       \tl_if_empty:nTF { ##1 }
1123       {
1124         \pdfannot_dict_remove:nn
1125         {link/#2}
1126         { C }
1127       }
1128       {
1129         \__hyp_color_set:ne {hyp/color/#1border}{##1}
1130         \color_export:nVN
1131         {hyp/color/#1border}
1132         \g__hyp_bordercolormodel_str
1133         \l__hyp_tmpa_tl
1134         \pdfannot_dict_put:nne
1135         {link/#2}
1136         { C }
1137         { [\l__hyp_tmpa_tl] }
1138       }
1139     }
1140   }
1141 }

1142 \keys_define:nn { hyp }
1143 {
1144   ,allbordercolors .meta:n =
1145   {
1146     ,linkbordercolor=#1
1147     ,urlbordercolor =#1
1148     ,filebordercolor=#1
1149     ,menubordercolor=#1
1150     ,runbordercolor =#1
1151   }
1152   ,allbordercolors .value_required:n = true
1153 }
1154 }
```

1155

11.3.2 Borderwidth and -arc

```

1156 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1157 {
1158   \keys_define:nn { hyp }
1159   {
1160     #1border .code:n =
1161     {
1162       \tl_if_empty:nTF { ##1 }
1163       {
1164         \pdfannot_dict_remove:nn
1165         {link/#2}
1166         { Border }
1167       }
1168       {
1169         \pdfannot_dict_put:nnn
1170         {link/#2}
1171         { Border }
1172         { [##1] }
1173       }
1174     }
1175   }
1176 }
1177 \keys_define:nn { hyp }
1178 {
1179   ,pdfborder .code:n =
1180   {
1181     \tl_if_empty:nTF { #1 }
1182     {
1183       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1184       {
1185         \pdfannot_dict_remove:nn
1186         {link/##2}
1187         { Border }
1188       }
1189     }
1190     {
1191       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1192       {
1193         \pdfannot_dict_put:nnn
1194         {link/##2}
1195         { Border }
1196         { [#1] }
1197       }
1198     }
1199   }
1200   ,pdfborder .initial:n = {0~0~1},
1201 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1202 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1203 {
runborderstyle (setup key) 1204 \keys_define:nn { hyp }
fileborderstyle (setup key) 1205 {
menuborderstyle (setup key) 1206 #1borderstyle .code:n =
1207 {
1208 \tl_if_empty:nTF { ##1 }
1209 {
1210 \pdfannot_dict_remove:nn
1211 {link/#2}
1212 { BS }
1213 }
1214 {
1215 \pdfannot_dict_put:nnn
1216 {link/#2}
1217 { BS }
1218 { <<##1>> }
1219 }
1220 }
1221 }
1222 }
1223 \keys_define:nn { hyp }
1224 {
1225 ,pdfborderstyle .code:n =
1226 {
1227 \tl_if_empty:nTF { #1 }
1228 {
1229 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1230 {
1231 \pdfannot_dict_remove:nn
1232 {link/##2}
1233 { BS }
1234 }
1235 }
1236 {
1237 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1238 {
1239 \pdfannot_dict_put:nnn
1240 {link/##2}
1241 { BS }
1242 { <<#1>> }
1243 }
1244 }
1245 }
1246 ,pdfborderstyle .initial:n = {},
1247 }

```

11.4 ocgcolorlinks

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

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

```

1248 \cs_new_protected:Npn \_hyp_ocg_init:
1249 {
1250   \pdf_object_new:n { \_hyp/OCG/View }
1251   \pdf_object_new:n { \_hyp/OCG/Print }
1252   \pdf_object_new:n { \_hyp/OCG/config }
1253   \pdf_object_new:n { \_hyp/OCG/refarray }
1254   \pdf_object_write:nne { \_hyp/OCG/refarray } { array }
1255   {
1256     \pdf_object_ref:n { \_hyp/OCG/View }
1257     \c_space_tl
1258     \pdf_object_ref:n { \_hyp/OCG/Print }
1259   }
1260   \pdf_object_write:nnn { \_hyp/OCG/View } { dict }
1261   {
1262     /Type/OCG
1263     /Name(View)
1264     /Usage
1265     <<
1266       /Print <</PrintState/OFF>>~
1267       /View <</ViewState/ON >>~
1268     >>
1269   }
1270   \pdf_object_write:nnn { \_hyp/OCG/Print } { dict }
1271   {
1272     /Type/OCG
1273     /Name(Print)
1274     /Usage
1275     <<
1276       /Print <</PrintState/ON>>~
1277       /View <</ViewState/OFF>>~
1278     >>
1279   }
1280   \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n { \_hyp/OCG/View } }
1281   \pdfmanagement_add:nne { Catalog / OCGProperties }{OCGs }{ \pdf_object_ref:n { \_hyp/OCG/Print } }
1282   \pdf_object_write:nne { \_hyp/OCG/config } { dict }
1283   {
1284     /OFF[\pdf_object_ref:n { \_hyp/OCG/Print } ]
1285     /AS[
1286       <<
1287         /Event/View
1288         /OCGs\c_space_tl \pdf_object_ref:n { \_hyp/OCG/refarray }
1289         /Category[/View]
1290       >>
1291       <<
1292         /Event/Print
1293         /OCGs\c_space_tl \pdf_object_ref:n { \_hyp/OCG/refarray }
1294         /Category[/Print]
1295       >>
1296       <<
1297         /Event/Export
1298         /OCGs\c_space_tl \pdf_object_ref:n { \_hyp/OCG/refarray }
1299         /Category[/Print]

```

```

1300         >>
1301     ]
1302 }
1303 \pdfmanagement_add:nne { Catalog / OCGProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1304 \cs_gset:Npn \__hyp_ocg_init: {}
1305 }

```

(End of definition for __hyp_ocg_init:.)

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

```

1306 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1307 {
1308     \hook_gput_code:nnn
1309     {pdfannot/link/#2/begin}
1310     {hyp/ocg}
1311     {
1312         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1313         {
1314             \__hyp_ocg_init:
1315             \group_begin:
1316             \hbox_set:Nw \l__hyp_tmpa_box
1317         }
1318     }
1319     \hook_gput_code:nnn
1320     {pdfannot/link/#2/end}
1321     {hyp/ocg}
1322     {
1323         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1324         {
1325             \hbox_set_end:
1326             \mbox
1327             {
1328                 \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1329                 \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1330                 \pdf_emc:
1331                 \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1332                 \group_begin:
1333                 \color_select:n { hyp/color/#1 }
1334                 \box_use_drop:N \l__hyp_tmpa_box
1335                 \group_end:
1336                 \pdf_emc:
1337             }
1338             \group_end:
1339         }
1340     }
1341 }

```

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

```

ocgcolorlink (setup key) 1342 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1343 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1344 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1345 {
ocgcolorrun (setup key) 1346     \keys_define:nn { hyp }
1347     {

```



```

1348     ,_ocgcolorlinks .meta:n =
1349     {
1350         ocgcolorlink=#1,
1351         ocgcolorurl=#1,
1352         ocgcolorfile=#1,
1353         ocgcolorrun=#1,
1354         ocgcolormenu=#1
1355     }
1356     ,_ocgcolorlinks .default:n = true
1357 }
1358 }
1359 {
1360     \keys_define:nn { hyp }
1361     {
1362         ,_ocgcolorlinks .code:n =
1363         {
1364             \msg_warning:nnee
1365             { hyp }
1366             { ignore-deprecated-or-unknown-option-in-pdf-version }
1367             { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1368         }
1369     }
1370 }
1371
1372 \keys_define:nn { hyp }
1373 {
1374     ,ocgcolorlinks .choice:
1375     ,ocgcolorlinks / true .meta:n =
1376     {
1377         pdfborder      = {0~0~0},
1378         pdfborderstyle = {},
1379         colorlinks     = false,
1380         _ocgcolorlinks = true
1381     }
1382     ,ocgcolorlinks / false .meta:n =
1383     {
1384         _ocgcolorlinks = false
1385     }
1386     ,ocgcolorlinks .default:n = {true}
1387 }
1388
1389 \seq_map_inline:Nn \c__hyp_annot_types_seq
1390 {
1391     \bool_lazy_or:nnTF
1392     { \pdf_version_compare_p:Nn > {1.4} }
1393     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1394     {
1395         \keys_define:nn { hyp }
1396         {
1397             ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1398         }
1399     }
1400     {
1401         \keys_define:nn { hyp }

```

```

1402         {
1403             ,ocgcolor#1 .code:n=
1404             {
1405                 \msg_warning:nnee
1406                 { hyp }
1407                 { ignore-deprecated-or-unknown-option-in-pdf-version }
1408                 { ocgcolor#1 }
1409                 { \pdf_version_major:.\pdf_version_minor: }
1410             }
1411         }
1412     }
1413 }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1414 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1415 {
1416     \keys_define:nn { hyp }
1417     {
1418         ,#1highlight .choices:nn =
1419         { /I, /N, /O, /P}
1420         {
1421             \pdfannot_dict_put:nnn
1422             {link/#2}
1423             { H }
1424             { ##1 }
1425         }
1426         ,#1highlight / .code:n =
1427         {
1428             \pdfannot_dict_remove:nn
1429             {link/#2}
1430             { H }
1431         }
1432         ,#1highlight / unknown .code:n =
1433         {
1434             \msg_warning:nnee { hyp } { unknown-choice+empty }
1435             { #1highlight }
1436             { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1437             { \exp_not:n {##1} }
1438         }
1439     }
1440 }
1441 }
1442 }
1443
1444
1445 \keys_define:nn { hyp }
1446 {
1447     ,pdfhighlight .choices:nn =
1448     { /I, /N, /O, /P}
1449     {
1450         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1451         {

```

```

1452         \pdfannot_dict_put:nnn
1453         {link/###2}
1454         { H }
1455         { #1 }
1456     }
1457 }
1458 ,pdfhighlight / .code:n =
1459 {
1460     \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1461     {
1462         \pdfannot_dict_remove:nn
1463         {link/##2}
1464         { H }
1465     }
1466 }
1467 ,pdfhighlight .initial:n = {/I},
1468 ,pdfhighlight / unknown .code:n =
1469 {
1470     \msg_warning:nnee { hyp } { unknown-choice+empty }
1471     { pdfhighlight }
1472     { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1473     { \exp_not:n {#1} }
1474 }
1475 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1476 \keys_define:nn { hyp }
hideurl (setup key) 1477 {
hidefile (setup key) 1478     hidelinks .meta:n =
hiderun (setup key) 1479     {
hidemenu (setup key) 1480         ,colorlinks      = false
1481         ,ocgcolorlinks = false
1482         ,pdfborder      = { 0~0~0 }
1483         ,pdfborderstyle=
1484     }
1485 }
1486
1487 \seq_map_inline:Nn \c__hyp_annot_types_seq
1488 {
1489     \keys_define:nn { hyp }
1490     {
1491         hide#1 .meta:n =
1492         {
1493             ,color#1      = false
1494             ,ocgcolor#1   = false
1495             ,#1border     = { 0~0~0 }
1496             ,#1borderstyle =
1497         }
1498     }
1499 }

```

11.7 color schemes and settings

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

```
colorscheme (setup key)
1500 \keys_define:nn { hyp }
1501 {
1502   colorscheme .code:n =
1503   {
1504     \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1505     {
1506       \keys_set:nn { hyp }
1507       {
1508         ##1 = ##2
1509       }
1510     }
1511   }
1512 }
1513 \keys_set:nn { hyp } {colorscheme=phetype}
```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```
unicode (setup key)
pdfencoding (setup key)
pdfversion (setup key)
1514 \keys_define:nn { hyp }
1515 {
1516   ,unicode .code:n = {}
1517   ,pdfencoding .code:n = {}
1518   ,pdfversion .code:n =
1519   {
1520     \msg_warning:nn { hyp }{ pdfversion-disabled }
1521   }
1522 }
1523 %
```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```
verbose (setup key)
debug (setup key)
draft (setup key)
final (setup key)
1524 \keys_define:nn { hyp }
1525 {
1526   ,verbose .legacy_if_set:n = {Hy@verbose}
1527   ,debug .legacy_if_set:n = {Hy@verbose}
1528 }
1529 \keys_define:nn { hyp }
1530 {
1531   ,draft .code:n =
1532   {
1533     \Hy@drafttrue
```

```

1534         \PassOptionsToPackage{draft}{bookmark}
1535     }
1536     ,final .code:n =
1537     {
1538         \Hy@finaltrue
1539         \PassOptionsToPackage{final}{bookmark}
1540     }
1541 }

extension (setup key)
hypertextnames (setup key) 1542 \keys_define:nn { hyp }
naturalnames (setup key) 1543 {
    pageanchor (setup key) 1544 ,extension .tl_set:N = \XR@ext
    linktoc (setup key) 1545 ,extension .initial:n= pdf
    linktocpage (setup key) 1546 ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
    plainpages (setup key) 1547 ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1548 ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1549 ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1550 ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1551 ,plainpages .legacy_if_set:n = {Hy@plainpages}
1552 }
1553
1554 \keys_define:nn { hyp }
1555 {
1556 ,linktoc .choices:nn = { none, section, all, page }
1557 {
1558     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1559 }
1560 ,linktoc / unknown .code:n =
1561 {
1562     \msg_warning:nneee { hyp } { unknown-choice }
1563     { linktoc }
1564     { none, section, all, page }
1565     { \exp_not:n {#1} }
1566 }
1567 ,linktocpage .choice:
1568 ,linktocpage / true .meta:n = {linktoc=page}
1569 ,linktocpage / false .meta:n = {linktoc=section}
1570 ,linktocpage .default:n = true
1571 }
1572

link (setup key) This booleans allow to disable the link types.
url (setup key) 1573 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1574 {
menu (setup key) 1575 \keys_define:nn { hyp }
run (setup key) 1576 {
1577     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1578 }
1579 }

1580 \keys_define:nn { hyp }
1581 {
1582     ,baseurl .code:n =

```

```

1583 {
1584   \_hyp\_text\_pdfstring:ooN { #1 } {\_hyp\_text\_enc\_uri\_print\_tl} \_hyp\_tmpa\_tl
1585   \tl\_if\_empty:NTF \_hyp\_tmpa\_tl
1586   {
1587     \pdfmanagement\_remove:nn {Catalog} { URI }
1588   }
1589   {
1590     \pdfmanagement\_add:nne {Catalog} { URI }{ <</Base \_hyp\_tmpa\_tl>> }
1591   }
1592   \_hyp\_store\_metadata:nn {baseurl}{#1}
1593 }
1594 %only false does something ...
1595 ,bookmarks .choice:
1596 ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1597 ,bookmarks / true .code:n = {}
1598 ,bookmarks .default:n = {true}
1599 ,bookmarksnumbered .legacy\_if\_set:n = {Hy@bookmarksnumbered}
1600 ,bookmarksopen .legacy\_if\_set:n = {Hy@bookmarksopen}
1601 ,bookmarksopenlevel .tl\_set:N = \@bookmarksopenlevel
1602 ,bookmarkstype .tl\_set:N = \Hy@bookmarkstype
1603 ,pdfcenterwindow .choice:
1604 ,pdfcenterwindow / false .code:n =
1605 {
1606   \pdfmanagement\_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1607 }
1608 ,pdfcenterwindow / true .code:n =
1609 {
1610   \pdfmanagement\_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1611 }
1612 ,pdfcenterwindow / .code:n =
1613 {
1614   \pdfmanagement\_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1615 }
1616 ,pdfcenterwindow / unknown .code:n =
1617 {
1618   \msg\_warning:nnee { hyp } { no-bool }
1619   { pdfcenterwindow }
1620   { \exp\_not:n {#1} }
1621 }
1622 ,pdfcenterwindow .default:n = true
1623 ,pdfdirection .choice:
1624 ,pdfdirection / L2R .code:n =
1625 {
1626   \pdfmanagement\_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1627 }
1628 ,pdfdirection / R2L .code:n =
1629 {
1630   \pdfmanagement\_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1631 }
1632 ,pdfdirection / .code:n =
1633 {
1634   \pdfmanagement\_remove:nn {Catalog / ViewerPreferences } { Direction }
1635 }
1636 ,pdfdirection / unknown .code:n =

```

```

1637     {
1638         \msg_warning:nneee { hyp } { unknown-choice+empty }
1639         { pdfdirection }
1640         { L2R , R2L }
1641         { \exp_not:n {#1} }
1642     }
1643 ,pdfdisplaydoctitle .choice:
1644 ,pdfdisplaydoctitle / false .code:n =
1645 {
1646     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1647 }
1648 ,pdfdisplaydoctitle / true .code:n =
1649 {
1650     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1651 }
1652 ,pdfdisplaydoctitle .default:n = true
1653 ,pdfduplex .choices:nn =
1654 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1655 {
1656     \pdf_version_compare:NnTF > {1.6}
1657     {
1658         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1659         { PrintDuplex } { /#1 }
1660     }
1661     {
1662         \msg_warning:nnee
1663         {hyp}
1664         {ignore-deprecated-or-unknown-option-in-pdf-version}
1665         {pdfduplex}
1666         {\pdf_version:}
1667     }
1668 }%
1669 ,pdfduplex / .code:n =
1670 {
1671     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintDuplex }
1672 }
1673 ,pdfduplex / unknown .code:n =
1674 {
1675     \msg_warning:nneee { hyp } { unknown-choice+empty }
1676     { pdfduplex }
1677     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1678     { \exp_not:n {#1} }
1679 }
1680 ,pdffitwindow .choice:
1681 ,pdffitwindow / false .code:n =
1682 {
1683     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1684 }
1685 ,pdffitwindow / true .code:n =
1686 {
1687     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1688 }
1689 ,pdffitwindow / .code:n =
1690 {

```

```

1691     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1692   }
1693   ,pdffitwindow .default:n = true
1694   ,pdffitwindow / unknown .code:n =
1695   {
1696     \msg_warning:nnee { hyp } { no-bool }
1697     { pdffitwindow }
1698     { \exp_not:n {#1} }
1699   }
1700   ,pdfmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1701   ,pdfmargin .initial:n = {1pt}
1702   ,pdfmenubar .choice:
1703   ,pdfmenubar / true .code:n =
1704   {
1705     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1706   }
1707   ,pdfmenubar / false .code:n =
1708   {
1709     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1710     { HideMenubar } { true }
1711   }
1712   ,pdfmenubar / .code:n =
1713   {
1714     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1715   }
1716   ,pdfmenubar .default:n = true
1717   ,pdfmenubar / unknown .code:n =
1718   {
1719     \msg_warning:nnee { hyp } { no-bool }
1720     { pdfmenubar }
1721     { \exp_not:n {#1} }
1722   }
1723   ,pdfnewwindow .choice:
1724   ,pdfnewwindow / true .code:n =
1725   {
1726     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1727     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1728   }
1729   ,pdfnewwindow / false .code:n =
1730   {
1731     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1732     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1733   }
1734   ,pdfnewwindow / .code:n =
1735   {
1736     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1737     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1738   }
1739   ,pdfnonfullscreenpagemode .choices:nn =
1740   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1741   {
1742     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1743     { NonFullScreenPageMode} {/#1}
1744   }

```



```

1745 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1746 {
1747     \pdf_version_compare:NnTF < {1.6}
1748     {
1749         %message
1750     }
1751     {
1752         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1753         {NonFullScreenPageMode}{/UseAttachments}
1754     }
1755 }
1756 ,pdfnonfullscreenpagemode / .code:n =
1757 {
1758     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1759 }
1760 ,pdfnonfullscreenpagemode / unknown .code:n =
1761 {
1762     \msg_warning:nnee { hyp } { unknown-choice+empty }
1763     { pdfnonfullscreenpagemode }
1764     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1765     { \exp_not:n {#1} }
1766 }
1767 ,pdfnumcopies .code:n =
1768 {
1769     \pdf_version_compare:NnTF > {1.6}
1770     {
1771         \tl_if_empty:nTF {#1}
1772         {
1773             \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1774         }
1775         {
1776             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1777             {NumCopies}{#1}
1778         }
1779     }
1780     {
1781         \msg_warning:nnee
1782         {hyp}
1783         {ignore-deprecated-or-unknown-option-in-pdf-version}
1784         {pdfnumcopies}
1785         {\pdf_version:}
1786     }
1787 }
1788 ,pdfpagelayout .choices:nn =
1789 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1790 { \pdfmanagement_add:nne {Catalog} { PageLayout }{/#1 } }
1791 ,pdfpagelayout / .code:n =
1792 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1793 ,pdfpagelayout / unknown .code:n =
1794 {
1795     \msg_warning:nnee { hyp } { unknown-choice+empty }
1796     { pdfpagelayout }
1797     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1798     { \exp_not:n {#1} }

```

```

1799     }
1800 ,pdfpagemode .choices:nn =
1801   { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1802   { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1803 ,pdfpagemode / UseAttachments .code:n =
1804   {
1805     \pdf_version_compare:NnTF > {1.5}
1806     {
1807       \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1808     }
1809     {
1810       \msg_warning:nnee
1811         {hyp}
1812         {ignore-deprecated-or-unknown-value-in-pdf-version}
1813         {UseAttachments}
1814         {\pdf_version:}
1815     }
1816   }
1817 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1818 ,pdfpagemode / unknown .code:n =
1819   {
1820     \msg_warning:nneee { hyp } { unknown-choice+empty }
1821     { pdfpagemode }
1822     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1823     { \exp_not:n {#1} }
1824   }
1825 ,pdfpagescrop .code:n =
1826   {
1827     \tl_if_empty:nTF {#1} %or blank?
1828     {
1829       \pdfmanagement_remove:nn {Pages} { CropBox }
1830     }
1831     {
1832       \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1833     }
1834   }
1835 ,pdfpicktraybypdfsize .choice:
1836 ,pdfpicktraybypdfsize / true .code:n =
1837   {
1838     \pdf_version_compare:NnTF > {1.6}
1839     {
1840       \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1841         { PickTrayByPDFSize } { true }
1842     }
1843     {
1844       \msg_warning:nnee
1845         {hyp}
1846         {ignore-deprecated-or-unknown-option-in-pdf-version}
1847         {pdfpicktraybypdfsize}
1848         {\pdf_version:}
1849     }
1850   }
1851 ,pdfpicktraybypdfsize / false .code:n =
1852   {

```

```

1853     \pdf_version_compare:NnTF > {1.6}
1854     {
1855         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1856         { PickTrayByPDFSize } { false }
1857     }
1858     {
1859         \msg_warning:nnee
1860         {hyp}
1861         {ignore-deprecated-or-unknown-option-in-pdf-version}
1862         {pdfpicktraybypdfsize}
1863         {\pdf_version:}
1864     }
1865 }
1866 ,pdfpicktraybypdfsize / .code:n =
1867 {
1868     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1869 }
1870 ,pdfpicktraybypdfsize / unknown .code:n =
1871 {
1872     \msg_warning:nnee { hyp } { no-bool }
1873     { picktraybypdfsize }
1874     { \exp_not:n {#1} }
1875 }
1876 ,pdfprintarea .choices:nn =
1877 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1878 {
1879     \pdf_version_compare:NnTF < {2.0}
1880     {
1881         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1882         { PrintArea } { /#1 }
1883     }
1884     {
1885         \msg_warning:nnee
1886         {hyp}
1887         {ignore-deprecated-or-unknown-option-in-pdf-version}
1888         {pdfprintarea}
1889         {\pdf_version:}
1890     }
1891 }%
1892 ,pdfprintarea / .code:n =
1893 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1894 ,pdfprintarea / unknown .code:n =
1895 {
1896     \msg_warning:nneee { hyp } { unknown-choice+empty }
1897     { pdfprintarea }
1898     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1899     { \exp_not:n {#1} }
1900 }
1901 ,pdfprintclip .choices:nn =
1902 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1903 {
1904     \pdf_version_compare:NnTF < {2.0}
1905     {
1906         \pdfmanagement_add:nne {Catalog / ViewerPreferences }

```

```

1907         { PrintClip } { /#1 }
1908     }
1909     {
1910         \msg_warning:nnee
1911         {hyp}
1912         {ignore-deprecated-or-unknown-option-in-pdf-version}
1913         {pdfprintclip}
1914         {\pdf_version:}
1915     }
1916 }%
1917 ,pdfprintclip / .code:n =
1918 {
1919     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1920 }
1921 ,pdfprintclip / unknown .code:n =
1922 {
1923     \msg_warning:nneee
1924     { hyp }
1925     { unknown-choice+empty }
1926     { pdfprintclip }
1927     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1928     { \exp_not:n {#1} }
1929 }
1930 ,pdfprintpagerange .code:n =
1931 {
1932     \pdf_version_compare:NnTF > {1.6}
1933     {
1934         \tl_if_empty:nTF { #1}
1935         {
1936             \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1937             { PrintPageRange }
1938         }
1939         {
1940             \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1941             {PrintPageRange}{[#1]}
1942         }
1943     }
1944     {
1945         \msg_warning:nnee
1946         {hyp}
1947         {ignore-deprecated-or-unknown-option-in-pdf-version}
1948         {pdfprintpagerange}
1949         {\pdf_version:}
1950     }
1951 }
1952 ,pdfprintscaling .choices:nn =
1953 { None, AppDefault }
1954 {
1955     \pdf_version_compare:NnTF > {1.5}
1956     {
1957         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1958         { PrintScaling } { /#1 }
1959     }
1960     {

```

```

1961         \msg_warning:nnee
1962         {hyp}
1963         {ignore-deprecated-or-unknown-option-in-pdf-version}
1964         {pdfprintscaling}
1965         {\pdf_version:}
1966     }
1967 }%
1968 ,pdfprintscaling / .code:n =
1969 {
1970     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1971 }
1972 ,pdfprintscaling / unknown .code:n =
1973 {
1974     \msg_warning:nnee { hyp } { unknown-choice+empty }
1975     { pdfprintarea }
1976     { None, AppDefault }
1977     { \exp_not:n {#1} }
1978 }
1979 ,pdfremotestartview .code:n =
1980 {
1981     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
1982     \exp_args:NNV
1983     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1984     {
1985         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1986     }
1987     {
1988         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1989         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1990     }
1991 }
1992 ,pdfremotestartview .initial:n = {Fit}
1993 % pdfstartpage is special as it shares code with pdfstartview
1994 ,pdfstartpage .code:n =
1995 {
1996     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1997     \bool_if:nTF
1998     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1999     {
2000         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2001     }
2002     {
2003         \pdfmanagement_add:nne {Catalog} { OpenAction }
2004         {
2005             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2006         }
2007     }
2008 }
2009 ,pdfstartpage .initial:n =1
2010 ,pdfstartview .code:n =
2011 {
2012     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
2013     \exp_args:NNV
2014     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_

```

```

2015     {
2016         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
2017     }
2018     {
2019         \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfstartview}
2020         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2021     }
2022     \bool_if:nTF
2023     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2024     {
2025         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2026     }
2027     {
2028         \pdfmanagement_add:nne {Catalog} { OpenAction }
2029         {
2030             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2031         }
2032     }
2033 }
2034 ,pdfstartview .initial:n = Fit
2035 ,pdftoolbar .choice:
2036 ,pdftoolbar / true .code:n =
2037 {
2038     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2039 }
2040 ,pdftoolbar / false .code:n =
2041 {
2042     \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2043     { HideToolbar } { true }
2044 }
2045 ,pdftoolbar / true .code:n =
2046 {
2047     \pdfmanagement_remove:nn {Catalog / ViewerPreferences} { HideToolbar }
2048 }
2049 ,pdftoolbar .default:n = true
2050 ,pdftoolbar / unknown .code:n =
2051 {
2052     \msg_warning:nnee { hyp } { no-bool }
2053     { pdftoolbar }
2054     { \exp_not:n {#1} }
2055 }
2056 % pdfview see below.
2057 ,pdfviewarea .choices:nn =
2058 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2059 {
2060     \pdf_version_compare:NnTF < {2.0}
2061     {
2062         \pdfmanagement_add:nnn {Catalog / ViewerPreferences}
2063         { ViewArea } { /#1 }
2064     }
2065     {
2066         \msg_warning:nnee
2067         {hyp}
2068         {ignore-deprecated-or-unknown-option-in-pdf-version}

```

```

2069         {pdfviewarea}
2070         {\pdf_version:}
2071     }
2072 }%
2073 ,pdfviewarea / .code:n =
2074 {
2075     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2076 }
2077 ,pdfviewarea / unknown .code:n =
2078 {
2079     \msg_warning:nneee { hyp } { unknown-choice+empty }
2080     { pdfviewarea }
2081     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2082     { \exp_not:n {#1} }
2083 }
2084 ,pdfviewclip .choices:nn =
2085 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2086 {
2087     \pdf_version_compare:NnTF < {2.0}
2088     {
2089         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2090         { ViewClip } { /#1 }
2091     }
2092     {
2093         \msg_warning:nnee
2094         {hyp}
2095         {ignore-deprecated-or-unknown-option-in-pdf-version}
2096         {pdfviewclip}
2097         {\pdf_version:}
2098     }
2099 }%
2100 ,pdfviewclip / .code:n =
2101 {
2102     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2103 }
2104 ,pdfviewclip / unknown .code:n =
2105 {
2106     \msg_warning:nneee { hyp } { unknown-choice+empty }
2107     { pdfviewclip }
2108     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2109     { \exp_not:n {#1} }
2110 }
2111 ,pdfwindowui .choice:
2112 ,pdfwindowui / true .code:n =
2113 {
2114     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2115 }
2116 ,pdfwindowui / false .code:n =
2117 {
2118     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2119     { HideWindowUI } { true }
2120 }
2121 ,pdfwindowui / .code:n =
2122 {

```

```

2123     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2124   }
2125   ,pdfwindowui / unknown .code:n =
2126   {
2127     \msg_warning:nnee { hyp } { no-bool }
2128     { pdfwindowui }
2129     { \exp_not:n {#1} }
2130   }
2131   ,pdfwindowui .default:n = true
2132 }

```

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

```

2133 \keys_define:nn { hyp }
2134 {
2135   ,pdfview .code:n =
2136   {
2137     \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2138     \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } } {
2139       {
2140         { xyz }
2141         {
2142           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2143           {
2144             \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2145             \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2146             {
2147               \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2148             }
2149             {
2150               \tl_set:Ne \l__hyp_dest_pdfview_tl
2151               {
2152                 \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2153               }
2154             }
2155           }
2156           {
2157             \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2158           }
2159         }
2160         { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2161         { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2162         { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2163         { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2164         { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2165         { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2166         { fitr }
2167         {
2168           \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2169           {
2170             \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2171           }
2172           {
2173             %ensure 4 values ...
2174             \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}

```



```

2175 \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2176 \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2177 \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2178 \hbox_set_to_wd:Nnn \l__hyp_dest_box
2179 {
2180   \fp_eval:n
2181   {
2182     round
2183     (
2184       abs
2185       (
2186         \seq_item:Nn\l__hyp_tmpa_seq{4}
2187         -
2188         (\seq_item:Nn\l__hyp_tmpa_seq{2})
2189       ),
2190       3
2191     )
2192   }bp
2193 }{}
2194 \box_set_dp:Nn \l__hyp_dest_box
2195 {
2196   \fp_eval:n
2197   {
2198     round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2199   }bp
2200 }
2201 \box_set_ht:Nn \l__hyp_dest_box
2202 {
2203   \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2204 }
2205 }
2206 }
2207 }
2208 {
2209   \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2210   \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2211 }
2212 }
2213 ,pdfview .initial:n = {xyz}
2214 }

```

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.

```

2215 \keys_define:nn { hyp }
2216 {
2217   ,pdflang .code:n =
2218   {
2219     \tl_if_empty:nF { #1 }
2220     {
2221       \pdfmanagement_add:nne {Catalog} { Lang } { { (#1) }

```

```

2222         \AddToDocumentProperties[document]{lang}{#1}
2223     }
2224 }
2225 }

```

12.3.1 “info keys”

pdfauthor (setup key) The keys store their value also in the metadata container, so that hyperxmp can use them.
 pdftitle (setup key) Creator and Producer can't be removed with the pdfmanagement, but we allow to set an
 pdfcreator (setup key) empty value. If the value begin with an optional argument, we assume a multilanguage
 pdfsubject (setup key) clist and use only the first value. The values are expanded with \text_expand:n
 pdfproducer (setup key) 2226 \regex_new:N\l__hyp_optlang_regex
 pdfkeywords (setup key) 2227 \regex_set:Nn\l__hyp_optlang_regex {\A\[[A-Za-z\-\]\+)(.*)}

```

2228 \cs_generate_variant:Nn\clist_item:nn{on}
2229 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2230 {
2231     \keys_define:nn { hyp }
2232     {
2233         pdf#1 .code:n =
2234         {
2235             \tl_set:N\l__hyp_tmpa_tl {\text_expand:n{##1}}
2236             \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2237             \tl_if_empty:NTF \l__hyp_tmpa_tl
2238             {
2239                 \str_case:nnF { #1 }
2240                 {
2241                     {creator}
2242                     {
2243                         \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2244                         \pdfmanagement_add:nne {Info}{Creator}{()}
2245                     }
2246                     {producer}
2247                     {
2248                         \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2249                         \pdfmanagement_add:nne {Info}{Producer}{()}
2250                     }
2251                 }
2252                 {
2253                     \pdfmanagement_remove:nn {Info}{#2}
2254                 }
2255             }
2256         {
2257             \tl_set:N\l__hyp_tmpb_tl {\clist_item:on{\l__hyp_tmpa_tl}{1}}
2258             \exp_args:NNV
2259             \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_s
2260             \seq_if_empty:NTF\l__hyp_tmpa_seq
2261             {
2262                 \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2263             }
2264             {
2265                 \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2266             }
2267             \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}

```

```

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

```

pdfcreationdate (setup key) These keys are not really needed. We store them too in the container. CreationDate and

pdfmoddate (setup key) ModDate should not use the hex encoding.

```

pdfmetadate (setup key) 2311 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2312 {
2313     \keys_define:nn { hyp }
2314     {
2315         pdf#1 .code:n =
2316         {
2317             \tl_if_blank:nTF {##1}
2318             {

```

```

2319         \pdfmanagement_remove:nn {Info}{#2}
2320     }
2321     {
2322         \pdfmanagement_add:nne {Info}{#2}{(##1)}
2323     }
2324     \__hyp_store_metadata:nn {pdf#1}{##1}
2325     \AddToDocumentProperties[document]{#1}{##1}
2326 }
2327 }
2328 \keys_define:nn { hyp / info }
2329 {
2330     #2 .code:n =
2331     {
2332         \tl_if_blank:nTF {##1}
2333         {
2334             \pdfmanagement_remove:nn {Info}{#2}
2335         }
2336         {
2337             \pdfmanagement_add:nne {Info}{#2}{(##1)}
2338         }
2339         \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}
2340     }
2341 }
2342 }
2343
2344 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2345 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2346 \keys_define:nn { hyp }
2347 {
2348     pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2349 }

```

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

```

2350 \keys_define:nn { hyp }
2351 {
2352     ,pdftrapped .code:n =
2353     {
2354         \exp_args:Nne
2355         \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2356     }
2357     ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2358     {
2359         \pdfmanagement_add:nne {Info}{Trapped}
2360         {/
2361             \str_uppercase:f { \str_head:n { #1 } }
2362             \str_lowercase:f { \str_tail:n { #1 } }
2363         }
2364         \__hyp_store_metadata:ne {pdftrapped}
2365         {
2366             \str_uppercase:f { \str_head:n { #1 } }
2367             \str_lowercase:f { \str_tail:n { #1 } }
2368         }
2369     }
2370     ,_pdftrapped / unknown .code:n =

```

```

2371     {
2372         \msg_warning:nneee { hyp } { unknown-choice }
2373         { pdftrapped }
2374         { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2375         { \exp_not:n {#1} }
2376     }
2377 }

```

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

```

2378 \keys_define:nn { hyp }
2379 {
2380     pdfinfo .code:n =
2381     {
2382         \keys_set:nn { hyp / info } { #1 }
2383     }
2384 }

```

Now we set some default values

```

2385 \keys_set:nn { hyp } {pdfcreator = LaTeX-with-hyperref}
2386 \keys_set:nn { hyp } {pdfauthor  = }
2387 \keys_set:nn { hyp } {pdftitle   = }
2388 \keys_set:nn { hyp } {pdfsubject = }

```

12.4 hyperxmp keys

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

```

2389 \clist_map_inline:nn
2390 {
2391     ,pdfcopyright
2392     ,pdftype
2393     ,pdflicenseurl
2394     ,pdfauthortitle
2395     ,pdfcaptionwriter
2396     ,pdfmetalang
2397     ,pdfsource
2398     ,pdfdocumentid
2399     ,pdfinstanceid
2400     ,pdfversionid
2401     ,pdfrendition
2402     ,pdfpublication
2403     ,pdfpubtype
2404     ,pdfbytes
2405     ,pdfnumpages
2406     ,pdfissn
2407     ,pdfeissn
2408     ,pdfisbn
2409     ,pdfbookedition
2410     ,pdfpublisher
2411     ,pdfvolumenum
2412     ,pdfissuenum
2413     ,pdfpagerange
2414     ,pdfdoi

```

```

2415 ,pdfurl
2416 ,pdfidentifier
2417 ,pdfsubtitle
2418 ,pdfpubstatus
2419 ,pdfcontactaddress
2420 ,pdfcontactcity
2421 ,pdfcontactregion
2422 ,pdfcontactpostcode
2423 ,pdfcontactcountry
2424 ,pdfcontactphone
2425 ,pdfcontactemail
2426 ,pdfcontacturl
2427 ,pdfdate
2428 }
2429 {
2430   \keys_define:nn { hyp }
2431   {
2432     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2433   }
2434 }
2435

```

12.5 Transitions

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

```

2436 \keys_define:nn { hyp }
2437 {
2438   pdfpageduration .code:n =
2439   {
2440     \tl_if_blank:nTF { #1 }
2441     {
2442       \pdfmanagement_remove:nn {Page}{Dur}
2443     }
2444     {
2445       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2446     }
2447   }
2448 }

```

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

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

The keys allowed in the argument of `pdfpagetransition` are

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

```

2449 \keys_define:nn { hyp }
2450 {
2451   pdfpagetransition .code:n =
2452   {
2453     \tl_if_blank:nTF {#1}
2454     {
2455       \pdfmanagement_remove:nn {Page}{Trans}
2456     }
2457     {
2458       \group_begin:
2459       \keys_set:nn { hyp / trans }{style=R,#1}
2460       \pdf_object_unnamed_write:ne { dict }
2461       {
2462         \pdfdict_use:n {l__hyp_page/Trans}
2463       }
2464       \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2465       \group_end:
2466     }
2467   }
2468 }
2469 \keys_define:nn { hyp / trans }
2470 {
2471   ,style .choices:nn =
2472   {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2473   { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2474   ,style / unknown .code:n =
2475   {
2476     \msg_warning:nneee { hyp } { unknown-choice }
2477     { trans / style }
2478     { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2479     { \exp_not:n {#1} }
2480   }
2481   ,duration .code:n =
2482   {
2483     \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
  
```

```

2484     }
2485     ,direction .choices:nn =
2486     {H,V}
2487     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2488     ,direction .choices:nn =
2489     {0,90,180,270,315}
2490     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2491     ,direction / None .code:n =
2492     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2493     ,direction / unknown .code:n =
2494     {
2495         \msg_warning:nneee { hyp } { unknown-choice }
2496         { trans / direction }
2497         {
2498             H~(horizontal,~only~Split,~Blinds),
2499             V~(vertical,~only~Split,~Blinds),
2500             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2501             90~(bottom~to~top,~only~Wipe),
2502             180~(right~to~left,~only~Wipe),
2503             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2504             315~(top~left~to~bottom,~only~Glitter),
2505             None~(only~Fly)
2506         }
2507         { \exp_not:n {#1} }
2508     }
2509     ,motion .choices:nn =
2510     {I,O}
2511     { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2512     ,motion / unknown .code:n =
2513     {
2514         \msg_warning:nneee { hyp } { unknown-choice }
2515         { trans / motion }
2516         { I~(inwards) , O~(outwards) }
2517         { \exp_not:n {#1} }
2518     }
2519     ,scale .code:n =
2520     { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2521     ,opaque .choices:nn = {true,false}
2522     { \pdfdict_put:nnn { l__hyp_page/Trans }{ B }{ #1 } }
2523     ,opaque / unknown .code:n =
2524     {
2525         \msg_warning:nneee { hyp } { unknown-choice }
2526         { trans / B }
2527         { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2528         { \exp_not:n {#1} }
2529     }
2530     % try to set unknown keys as style
2531     ,unknown .code:n =
2532     {
2533         % warning ...
2534         \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2535     }
2536 }

```


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

```

2537 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
    Unfinished Form field code
2538 \NewDocumentCommand \MakeFieldObject { m m }
2539 {
2540   \pdfxform_new:nnn { #2 }{ } { #1 }
2541 }
2542
2543
2544 \prop_new:N    \g__hyp_AcroForm_CoFields_prop
2545 \prop_new:N    \g__hyp_AcroForm_Fields_prop
2546
2547 \let\HyField@afields\@empty
2548 \let\HyField@cofields\@empty
2549 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2550
2551 % the value doesn't matter, but with a prop we avoid duplicates and it is
2552 % clearly faster than removing them from a sequence
2553 \def\HyField@AuxAddToFields#1
2554 {
2555   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2556 }%
2557
2558 %fields with empty key get a value too -- lets hope that
2559 %this give the expected behaviour
2560 \def\HyField@AuxAddToCoFields #1 #2
2561 {
2562   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2563 }
2564
2565 \Hy@AtBeginDocument
2566 {
2567   \if@filesw
2568     \immediate\write\@mainaux{%
2569       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2570     }%
2571     \immediate\write\@mainaux{%
2572       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2573     }%
2574   \fi
2575   \let\HyField@AfterAuxOpen\@firstofone
2576 }%
2577
2578 \def\HyField@AddToFields
2579 {
2580   \exp_args:Ne\HyField__hypAddToFields
2581   {
2582     \pdfannot_box_ref_last:
2583   }
2584   \ifx\Fld@calculate@code\@empty
2585   \else
2586     \begingroup
2587     \Hy@safe@activetrue

```

```

2588         \edef\Hy@temp{%
2589             \endgroup
2590             \if@filesw
2591                 \write\@mainaux
2592                 {
2593                     \string\HyField@AuxAddToCoFields
2594                     {
2595                         \Fld@calculate@sortkey
2596                     }
2597                     {
2598                         \pdfannot_box_ref_last:
2599                     }
2600                 }
2601             \fi
2602         }%
2603     \Hy@temp
2604 \fi
2605 }%
2606
2607 \def\HyField__hypAddToFields#1{
2608     \HyField@AfterAuxOpen{%
2609         \if@filesw
2610             \write\@mainaux{%
2611                 \string\HyField@AuxAddToFields{#1}%
2612             }%
2613         \fi
2614     }%
2615 }%
2616
2617 \ExplSyntaxOff
2618 \ExplSyntaxOn
2619
2620 \def\@Form[#1]
2621 {
2622     \kvsetkeys{Form}{#1}
2623     \pdf@ifdraftmode{}
2624     {
2625         \Hy@FormObjects
2626         \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2627         {
2628             \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2629             %\pdfmanagement_show:n { Catalog / AcroForm }
2630         }
2631         \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2632         {
2633             \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2634             {
2635                 \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2636             }
2637             \seq_sort:Nn \l__hyp_tmpa_seq
2638             {
2639                 \str_compare:nNnTF {##1} > {##2}
2640                 { \sort_return_swapped: }
2641                 { \sort_return_same: }

```

```

2642     }
2643     \seq_map_inline:Nn \l__hyp_tmpa_seq
2644     {
2645         \pdfmanagement_add:nne { Catalog / AcroForm }
2646         { CO }
2647         {
2648             \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2649         }
2650     }
2651 }
2652 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2653 {ZaDb} {\pdf_object_ref:n {\__hyp/Font/ZaDb} }
2654 \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2655 {Helv} {\pdf_object_ref:n {\__hyp/Font/Helv} }
2656 \pdfmanagement_add:nne {Catalog /AcroForm}
2657 {DA}{(/Helv~10~Tf~0~g)}
2658 \pdfmeta_standard_verify:nTF {form_no_NeedAppearances}
2659 {
2660     \legacy_if:nT { HyField@NeedAppearances }
2661     {
2662         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2663     }
2664 }
2665 {
2666     \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2667 }
2668 }
2669 }
2670 \ExplSyntaxOff
2671 \let\@endForm\@empty
2672 \let\HyAnn@AbsPageLabel\@empty
2673 \let\Fld@pageobjref\@empty
2674
2675 \ExplSyntaxOn
2676 \newcount\HyAnn@Count
2677 \HyAnn@Count=\z@
2678 \def\HyAnn@AbsPageLabel
2679 {
2680     \global\advance\HyAnn@Count by\@ne
2681     %\zref@labelbyprops{HyAnn@\the\HyAnn@Count}{abspage}%
2682     %\zref@labelbylist {HyAnn@\the\HyAnn@Count} {l3pdf}
2683     %\zref@refused{HyAnn@\the\HyAnn@Count}%
2684     \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2685     \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2686 }%
2687 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2688 \def\Fld@pageobjref
2689 {
2690     \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2691     {
2692         /P~\pdf_pageobject_ref:e
2693         {
2694             \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2695         }

```

```

2696     }
2697 }
2698 \ExplSyntaxOff
2699 \ExplSyntaxOn
2700 %% check if the attr should be set through
2701 %% hooks.
2702 %% check if options are missing.
2703 \def\@TextField[#1]#2{% parameters, label
2704   \def\Fld@name{#2}%
2705   \let\Fld@default\@empty
2706   \let\Fld@value\@empty
2707   \def\Fld@width{\DefaultWidthofText}%
2708   \def\Fld@height{%
2709     \ifFld@multiline
2710       \DefaultHeightofTextMultiline
2711     \else
2712       \DefaultHeightofText
2713     \fi
2714   }%
2715   \begingroup
2716     \expandafter\HyField@SetKeys\expandafter{%
2717       \DefaultOptionsofText,#1%
2718     }%
2719     \PDFForm@Name
2720     \HyField@FlagsText
2721     \ifFld@hidden\def\Fld@width{1sp}\fi
2722     \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2723     \LayoutTextField{#2}{%
2724       \leavevmode
2725       \HyAnn@AbsPageLabel
2726       \Hy@escapeform\PDFForm@Text
2727       \pdfannot_box:nnnn
2728         {\Fld@width}
2729         {\Fld@height}
2730         {Opt} %is this correct?
2731         {\PDFForm@Text}
2732       \MakeTextField{\Fld@width}{\Fld@height}
2733       \HyField@AddToFields
2734     }%
2735   \endgroup
2736 }
2737 \providecommand\@curropt{}
2738 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2739   \def\Fld@name{#2}
2740   \let\Fld@default\relax
2741   \let\Fld@value\relax
2742   \def\Fld@width{\DefaultWidthofChoiceMenu}
2743   \def\Fld@height{\DefaultHeightofChoiceMenu}
2744   \begingroup
2745     \Fld@menulength=0 %
2746     \@tempdima\z@
2747     \clist_map_variable:nNn { #3 } \@curropt
2748     %\@for\@curropt:=#3\do
2749     {%

```

```

2750 \expandafter\Fld@checkequals\@curropt==\\%
2751 \Hy@StepCount\Fld@menulength
2752 \settowidth{\@tempdimb}{\@currDisplay}%
2753 \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2754 }%
2755 \advance\@tempdima by~15\p@
2756 \begingroup
2757 \HyField@SetKeys{#1}
2758 \edef\x{\endgroup
2759 \noexpand\expandafter
2760 \noexpand\HyField@SetKeys
2761 \noexpand\expandafter{%
2762 \expandafter\noexpand\csname DefaultOptionsof%
2763 \ifFld@radio
2764 Radio%
2765 \else
2766 \ifFld@combo
2767 \ifFld@popdown
2768 PopdownBox%
2769 \else
2770 ComboBox%
2771 \fi
2772 \else
2773 ListBox%
2774 \fi
2775 \fi
2776 \endcsname
2777 }%
2778 }\x
2779 \HyField@SetKeys{#1}%
2780 \PDFForm@Name
2781 \ifFld@hidden\def\Fld@width{1sp}\fi
2782 \ifx\Fld@value\relax
2783 \let\Fld@value\Fld@default
2784 \fi
2785 \LayoutChoiceField{#2}{%
2786 \ifFld@radio
2787 \HyField@FlagsRadioButton
2788 \__hypRadio{#3}%
2789 \else
2790 \begingroup
2791 \HyField@FlagsChoice
2792 \ifdim\Fld@width<\@tempdima
2793 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2794 \edef\Fld@width{\the\@tempdima}%
2795 \fi
2796 \ifFld@combo
2797 \else
2798 \@tempdima=\the\Fld@menulength\Fld@charsize
2799 \advance\@tempdima by~\Fld@borderwidth bp %
2800 \advance\@tempdima by~\Fld@borderwidth bp %
2801 \edef\Fld@height{\the\@tempdima}%
2802 \fi
2803 \__hypListbox{#3}%

```

```

2804         \endgroup
2805     \fi
2806 }%
2807 \endgroup
2808 }
2809
2810 \def\__hypRadio#1{%
2811     \Fld@listcount=0~%
2812     %\show\Fld@default
2813     \EdefEscapeName\Fld@default{\Fld@default}%
2814     \clist_map_variable:nNn { #1 } \@curropt
2815     %\@for\@curropt:=#1\do
2816     {%
2817         \expandafter\Fld@checkequals\@curropt==\%
2818         \EdefEscapeName\@currValue{\@curropt}%
2819         \Hy@StepCount\Fld@listcount
2820         \@currDisplay\space
2821         \leavevmode
2822         \HyAnn@AbsPageLabel
2823         \Hy@escapeform\PDFForm@Radio
2824         \pdfxform_if_exist:nF { __hyp_xform_Ding }
2825         {
2826             \pdfxform_new:nnn { __hyp_xform_Ding } {}
2827             {
2828                 \group_begin:
2829                 \fontfamily{pzd}
2830                 \fontencoding{U}
2831                 \fontseries{m}
2832                 \fontshape{n}
2833                 \selectfont
2834                 \char123
2835                 \group_end:
2836             }
2837         }
2838         \pdfannot_box:nnne
2839         {\Fld@width}
2840         {\Fld@height}
2841         {0pt} %is this correct?
2842         {
2843             \PDFForm@Radio
2844             /AP
2845             <<
2846             /N
2847             <<
2848             /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2849             %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2850             >>
2851             >>
2852         }
2853         {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2854         \int_compare:nNnT { \Fld@listcount } = { 1 }
2855         { \HyField@AddToFields }
2856         \c_space_tl % deliberate space between radio buttons
2857         % to do: --> should be configurable

```

```

2858 }%
2859 }
2860
2861 \newcount\Fld@listcount
2862 \def\_hypListbox#1
2863 {
2864   \HyField@PDFChoices{#1}
2865   \mode_leave_vertical:
2866   \HyAnn@AbsPageLabel
2867   \Hy@escapeform\PDFForm@List
2868   \pdfannot_box:nnnn
2869   {\Fld@width}{\Fld@height}{0pt}
2870   {\PDFForm@List}
2871   \MakeChoiceField{\Fld@width}{\Fld@height}
2872   \HyField@AddToFields
2873 }
2874
2875
2876 \def\@PushButton[#1]#2{% parameters, label
2877   \def\Fld@name{#2}%
2878   \group_begin:
2879     \exp_args:No\HyField@SetKeys
2880     {
2881       \DefaultOptionsofPushButton,#1
2882     }
2883     \PDFForm@Name
2884     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2885     {
2886       \HyField@FlagsPushButton
2887       \legacy_if:nT {Fld@hidden}
2888       {
2889         \def\Fld@width{1sp}
2890       }
2891       \LayoutPushButtonField
2892       {
2893         \mode_leave_vertical:
2894         \HyAnn@AbsPageLabel
2895         \Hy@escapeform\PDFForm@Push
2896         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2897         \pdfannot_box:nnnn
2898         {\box_wd:N\l_tmpa_box}
2899         {\box_ht:N\l_tmpa_box}
2900         {\box_dp:N\l_tmpa_box} %is this correct?
2901         {\PDFForm@Push}
2902         {\box_use:N\l_tmpa_box}
2903         \HyField@AddToFields
2904       }
2905     }
2906     {
2907       \msg_error:nn { hyp }{ pdfa-no-push-button }
2908       \LayoutPushButtonField
2909       {
2910         \mode_leave_vertical:
2911         \MakeButtonField{#2}

```

```

2912     }
2913 }
2914 \group_end:
2915 }
2916
2917 \def\@Submit[#1]#2
2918 {
2919   \def\Fld@width {\DefaultWidthofSubmit}
2920   \def\Fld@height{\DefaultHeightofSubmit}
2921   \group_begin:
2922     \exp_args:No\HyField@SetKeys
2923     {
2924       \DefaultOptionsofSubmit,#1
2925     }
2926   \HyField@FlagsPushButton
2927   \HyField@FlagsSubmit
2928   \legacy_if:nT { Fld@hidden }
2929   {
2930     \def\Fld@width{1sp}
2931   }
2932   \mode_leave_vertical:
2933   \HyAnn@AbsPageLabel
2934   \Hy@escapeform\PDFForm@Submit
2935   \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2936   \pdfxform_if_exist:nF
2937   { __hyp_xform_Submit }
2938   {
2939     \pdfxform_new:nnn { __hyp_xform_Submit }{}
2940     {
2941       \fbox{\color_select:n{yellow}\textsf{Submit}}
2942     }
2943     \pdfxform_new:nnn { __hyp_xform_SubmitP }{}
2944     {
2945       \fbox{\color_select:n{yellow}\textsf{SubmitP}}
2946     }
2947   }
2948   \pdfannot_box:nnnn
2949   {\box_wd:N\l_tmpa_box}
2950   {\box_ht:N\l_tmpa_box}
2951   {\box_dp:N\l_tmpa_box} %is this correct?
2952   {
2953     \PDFForm@Submit
2954     /AP<<
2955       /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2956       /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2957     >>
2958   }
2959   \HyField@AddToFields
2960   \box_use:N\l_tmpa_box
2961
2962   \group_end:
2963 }
2964
2965 \def\@Reset[#1]#2

```



```

2966 {
2967   \def\Fld@width {\DefaultWidthofReset}
2968   \def\Fld@height{\DefaultHeightofReset}
2969   \group_begin:
2970     \exp_args:No\HyField@SetKeys
2971     {
2972       \DefaultOptionsofReset,#1
2973     }
2974     \mode_leave_vertical:
2975     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2976     {
2977       \HyField@FlagsPushButton
2978       \legacy_if:nT { Fld@hidden }
2979       { \def\Fld@width{1sp} }
2980       \HyAnn@AbsPageLabel
2981       \Hy@escapeform\PDFForm@Reset
2982       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2983       \pdfannot_box:nnnn
2984       {\box_wd:N\l_tmpa_box}
2985       {\box_ht:N\l_tmpa_box}
2986       {\box_dp:N\l_tmpa_box} %is this correct?
2987       { \PDFForm@Reset }
2988       \HyField@AddToFields
2989       \box_use:N \l_tmpa_box
2990     }
2991     {
2992       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2993       \MakeButtonField{#2}
2994     }
2995   \group_end:
2996 }
2997
2998 \def\@CheckBox[#1]#2
2999 {% parameters, label
3000   \def\Fld@name{#2}
3001   \def\Fld@default{0}
3002   \group_begin:
3003     \def\Fld@width {\DefaultWidthofCheckBox}
3004     \def\Fld@height{\DefaultHeightofCheckBox}
3005     \exp_args:No\HyField@SetKeys
3006     {
3007       \DefaultOptionsofCheckBox,#1
3008     }
3009     \PDFForm@Name
3010     \HyField@FlagsCheckBox
3011     \legacy_if:nT { Fld@hidden }
3012     {
3013       \def\Fld@width{1sp}
3014     }
3015     \LayoutCheckField{#2}
3016     {
3017       \mode_leave_vertical:
3018       \HyAnn@AbsPageLabel
3019       \Hy@escapeform\PDFForm@Check

```

```

3020 \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3021 {
3022   \pdfxform_new:nnn
3023   {__hyp_xform_CheckMarkYes}-{ }
3024   {
3025     \group_begin:
3026     \fontfamily{pzd}
3027     \fontencoding{U}
3028     \fontseries{m}
3029     \fontshape{n}
3030     \selectfont
3031     \char51
3032     \group_end:
3033   }
3034   \pdfxform_new:nnn
3035   {__hyp_xform_CheckMarkOff}-{ }
3036   {
3037     \group_begin:
3038     \fontfamily{pzd}
3039     \fontencoding{U}
3040     \fontseries{m}
3041     \fontshape{n}
3042     \selectfont
3043     \phantom{\char51} %perhaps xetex needs some small glyph ..
3044     \group_end:
3045   }
3046 }
3047 \pdfannot_box:nnnn
3048 {\Fld@width}
3049 {\Fld@height}
3050 {Opt} %is this correct?
3051 {\PDFForm@Check}
3052 \HyField@AddToFields %check if this works with xelatex ...
3053 }
3054 \group_end:
3055 }
3056 \ExplSyntaxOff
3057
3058 %hm. Should a luatex driver use type1 fonts in fields????
3059 \ExplSyntaxOn
3060 \def\Hy@FormObjects
3061 {
3062   \pdf_object_new:n {__hyp/Encoding/pdfdoc }
3063   \pdf_object_new:n {__hyp/Font/ZaDb }
3064   \pdf_object_new:n {__hyp/Font/Helv }
3065   \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3066   {
3067     /Type/Encoding
3068     /Differences[
3069       24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3070       /ring/tilde
3071       \c_space_tl
3072       39/quotesingle
3073       \c_space_tl

```

```

3074      96/grave %
3075      \iow_newline:
3076      128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3077      /fraction/guilsinglleft/guilsinglright/minus/perthousand
3078      /quotedblbase/quotedblleft/quotedblright/quoteleft
3079      /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3080      /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3081      \iow_newline:
3082      164/currency
3083      \c_space_tl
3084      166/brokenbar
3085      \c_space_tl
3086      168/dieresis/copyright/ordfeminine
3087      \c_space_tl
3088      172/logicalnot/.notdef/registered/macron/degree/plusminus
3089      /twosuperior/threesuperior/acute/mu
3090      \c_space_tl
3091      183/periodcentered/cedilla/onesuperior/ordmasculine
3092      \c_space_tl
3093      188/onequarter/onehalf/threequarters
3094      \iow_newline:
3095      192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3096      /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3097      /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3098      /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3099      /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3100      /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3101      /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3102      /iacute/icircumflex/idieresis/eth/ntilde/ograde/oacute
3103      /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3104      /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3105    ]
3106  }
3107  \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3108  {
3109    /Type/Font
3110    /Subtype/Type1
3111    /Name/ZaDb
3112    /BaseFont/ZapfDingbats
3113  }
3114  \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3115  {
3116    /Type/Font
3117    /Subtype/Type1
3118    /Name/Helv
3119    /BaseFont/Helvetica
3120    /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3121  }
3122  \global\let\Hy@FormObjects\relax
3123  }
3124  \ExplSyntaxOff
3125  \providecommand*{\Fld@pageobjref}{\}
3126  \ifcsname pdf@escapestring\endcsname
3127    \def\Hy@escapeform#1{%

```

```

3128     \ifHy@pdfescapeform
3129     \let\Hy@escapestring\pdfescapestring
3130     \else
3131     \let\Hy@escapestring\@firstofone
3132     \fi
3133   }%
3134   \Hy@escapeform{}}%
3135 \else
3136   \let\Hy@escapestring\@firstofone
3137   \def\Hy@escapeform#1{%
3138     \ifHy@pdfescapeform
3139     \def\Hy@escapestring##1{%
3140       \noexpand\Hy@escapestring{\noexpand##1}%
3141     }%
3142     \edef\Hy@temp{#1}%
3143     \expandafter\Hy__hypescapeform\Hy@temp\Hy@escapestring{}\@nil
3144     \def\Hy@escapestring##1{%
3145       \@ifundefined{Hy@esc@\string##1}{%
3146         ##1%
3147         \ThisShouldNotHappen
3148       }{%
3149         \csname Hy@esc@\string##1\endcsname
3150       }%
3151     }%
3152     \else
3153     \let\Hy@escapestring\@firstofone
3154     \fi
3155   }%
3156   \def\Hy__hypescapeform#1\Hy@escapestring#2#3\@nil{%
3157     \ifx\#3\%
3158     \else
3159       \expandafter
3160       \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3161       \Hy@ReturnAfterFi{%
3162         \Hy__hypescapeform#3\@nil
3163       }%
3164     \fi
3165   }%
3166 \fi
3167 \def\PDFForm@Name{%
3168   \PDFForm__hypName\Fld@name
3169   \ifx\Fld@altname\relax
3170   \else
3171     \PDFForm__hypName\Fld@altname
3172   \fi
3173   \ifx\Fld@mappingname\relax
3174   \else
3175     \PDFForm__hypName\Fld@mappingname
3176   \fi
3177 }
3178 \def\PDFForm__hypName#1{%
3179   \begingroup
3180   \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3181     \ifHy@unicode

```

```

3182         \Hy@unicodedefalse
3183     \fi
3184 \fi
3185     \pdfstringdef\Hy@gtemp#1%
3186 \endgroup
3187 \let#1\Hy@gtemp
3188 }
3189 \def\Fld@X@additionalactions{%
3190     \ifx\Fld@keystroke@code\@empty
3191     \else
3192         /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3193     \fi
3194     \ifx\Fld@format@code\@empty
3195     \else
3196         /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3197     \fi
3198     \ifx\Fld@validate@code\@empty
3199     \else
3200         /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3201     \fi
3202     \ifx\Fld@calculate@code\@empty
3203     \else
3204         /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3205     \fi
3206     \ifx\Fld@onfocus@code\@empty
3207     \else
3208         /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3209     \fi
3210     \ifx\Fld@onblur@code\@empty
3211     \else
3212         /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3213     \fi
3214     \ifx\Fld@onmousedown@code\@empty
3215     \else
3216         /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3217     \fi
3218     \ifx\Fld@onmouseup@code\@empty
3219     \else
3220         /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3221     \fi
3222     \ifx\Fld@onenter@code\@empty
3223     \else
3224         /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3225     \fi
3226     \ifx\Fld@onexit@code\@empty
3227     \else
3228         /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3229     \fi
3230 }
3231 \ExplSyntaxOn
3232 \def\Fld@additionalactions
3233 {%
3234     \exp_args:Ne\str_if_eq:nnF {\Fld@X@additionalactions}{ }
3235     {

```

```

3236     \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3237     {/AA<<\Fld@X@additionalactions>>}
3238   }
3239 }
3240 \ExplSyntaxOff
3241 \def\Fld@annotnames{%
3242   /T(\Fld@name)%
3243   \ifx\Fld@altname\relax
3244   \else
3245     /TU(\Fld@altname)%
3246   \fi
3247   \ifx\Fld@mappingname\relax
3248   \else
3249     /TM(\Fld@mappingname)%
3250   \fi
3251 }
3252 \ExplSyntaxOn
3253 \def\PDFForm@Check
3254 {
3255   /Subtype/Widget
3256   ~\Fld@annotflags
3257   ~\Fld@pageobjref
3258   ~\Fld@annotnames
3259   /FT/Btn
3260   \Fld@flags
3261   /Q~\Fld@align
3262   /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3263   /AP
3264   <<
3265     /N
3266     <<
3267       /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3268       /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3269     >>
3270   >>
3271   /MK<<
3272     \int_compare:nNnF {\Fld@rotation}={0}
3273     {
3274       /R~\Fld@rotation
3275     }
3276     \tl_if_empty:NF\Fld@bordercolor
3277     {
3278       /BC[\Fld@bordercolor]
3279     }
3280     \tl_if_empty:NF\Fld@bcolor
3281     {
3282       /BG[\Fld@bcolor]
3283     }
3284     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3285   >>
3286   /DA
3287   (
3288     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3289     \tl_if_empty:NF \Fld@color

```

```

3290         {
3291             \c_space_tl \Fld@color
3292         }
3293     )
3294 /H/P
3295 \legacy_if:nTF {Fld@checked}
3296 {
3297     /V/Yes /AS/Yes
3298 }
3299 {
3300     /V/Off /AS/Off
3301 }
3302 \Fld@additionalactions
3303 }
3304 \ExplSyntaxOff
3305 \ExplSyntaxOn
3306 \def\PDFForm@Push
3307 {
3308     /Subtype/Widget
3309     ~\Fld@annotflags
3310     ~\Fld@pageobjref
3311     ~\Fld@annotnames
3312     /FT/Btn
3313     ~\Fld@flags
3314     /H/P
3315     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3316     \bool_if:nT
3317     {
3318         !\int_compare_p:nNn {\Fld@rotation} = {0}
3319         ||
3320         \tl_if_exist_p:N \Fld@bordercolor
3321     }
3322     {
3323         /MK
3324         <<
3325         \int_compare:nNnF {\Fld@rotation} = {0}
3326         {
3327             /R~\Fld@rotation
3328         }
3329         \tl_if_exist:NT \Fld@bordercolor
3330         {
3331             /BC[\Fld@bordercolor]
3332         }
3333         >>
3334     }
3335     /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3336     \Fld@additionalactions
3337 }
3338
3339 \ExplSyntaxOff
3340 \def\PDFForm@List{%
3341     /Subtype/Widget%
3342     \Fld@annotflags
3343     \Fld@pageobjref

```

```

3344 \Fld@annotnames
3345 /FT/Ch%
3346 \Fld@flags
3347 /Q \Fld@align
3348 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3349 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3350 \ifx\Fld@bordercolor\relax\else 1\fi
3351 \ifx\Fld@bcolor\relax \else 1\fi
3352 \space
3353 \else
3354 /MK<<%
3355 \ifnum\Fld@rotation=\z@
3356 \else
3357 /R \Fld@rotation
3358 \fi
3359 \ifx\Fld@bordercolor\relax
3360 \else
3361 /BC[\Fld@bordercolor]%
3362 \fi
3363 \ifx\Fld@bcolor\relax
3364 \else
3365 /BG[\Fld@bcolor]%
3366 \fi
3367 >>%
3368 \fi
3369 /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3370 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3371 \Fld@choices
3372 \Fld@additionalactions
3373 }
3374 \ExplSyntaxOn
3375 \def\PDFForm@Radio
3376 {
3377 /Subtype/Widget
3378 ~\Fld@annotflags
3379 ~\Fld@pageobjref
3380 ~\Fld@annotnames
3381 /FT/Btn
3382 \Fld@flags
3383 /H/P
3384 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3385 /MK<<
3386 \ifnum\Fld@rotation=\z@
3387 \else
3388 /R~\Fld@rotation
3389 \fi
3390 \ifx\Fld@bordercolor\relax
3391 \else
3392 /BC[\Fld@bordercolor]%
3393 \fi
3394 \ifx\Fld@bcolor\relax
3395 \else
3396 /BG[\Fld@bcolor]%
3397 \fi

```



```

3398     /CA(\Hy@escapestring{\Fld@radiosymbol})%
3399     >>
3400     /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3401         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3402     \ifx\Fld@default\@empty
3403         /V/Off%
3404         /DV/Off%
3405     \else
3406         /V/\Fld@default
3407         /DV/\Fld@default
3408     \fi
3409     \Fld@additionalactions
3410 }
3411 \ExplSyntaxOff
3412 \ExplSyntaxOn
3413 % Does an appearance dict make sense here?
3414 \def\PDFForm@Text
3415 {
3416     /Subtype/Widget
3417     ~\Fld@annotflags
3418     ~\Fld@pageobjref
3419     ~\Fld@annotnames
3420     /FT/Tx
3421     ~\Fld@flags
3422     /Q~\Fld@align
3423     /BS<</W~\Fld@borderwidth\c_space_tl /S /\Fld@borderstyle>>
3424     \bool_if:nT
3425     {
3426         !\int_compare_p:nNn {\Fld@rotation} = {0}
3427         ||
3428         \tl_if_exist_p:N \Fld@bordercolor
3429         ||
3430         \tl_if_exist_p:N \Fld@bcolor
3431     }
3432     {
3433         /MK
3434         <<
3435         \int_compare:nNnF {\Fld@rotation} = {0}
3436         {
3437             /R~\Fld@rotation
3438         }
3439         \tl_if_exist:NT \Fld@bordercolor
3440         {
3441             /BC[\Fld@bordercolor]
3442         }
3443         \tl_if_exist:NT \Fld@bcolor
3444         {
3445             /BG[\Fld@bcolor]
3446         }
3447         >>
3448     }
3449     /DA
3450     (
3451         /Helv~\strip@pt\Fld@charsize\c_space_tl Tf

```

```

3452     \tl_if_empty:NF {\c_space_tl\Fld@color}
3453   )
3454   /DV(\Hy@escapestring{\Fld@default})
3455   /V(\Hy@escapestring{\Fld@value})
3456   ~\Fld@additionalactions
3457   \int_compare:nNnT { \Fld@maxlen}>{0}
3458   {
3459     /MaxLen~\Fld@maxlen
3460   }
3461 }
3462 \ExplSyntaxOff
3463
3464 \def\PDFForm@Submit{%
3465   /Subtype/Widget%
3466   \Fld@annotflags
3467   \Fld@pageobjref
3468   \Fld@annotnames
3469   /FT/Btn%
3470   \Fld@flags
3471   /H/P%
3472   /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3473   \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3474     \ifx\Fld@bordercolor\relax\else 1\fi
3475     \space
3476   \else
3477     /MK<<%
3478     \ifnum\Fld@rotation=\z@
3479     \else
3480       /R \Fld@rotation
3481     \fi
3482     \ifx\Fld@bordercolor\relax
3483     \else
3484       /BC[\Fld@bordercolor]%
3485     \fi
3486     >>%
3487   \fi
3488   /A<<%
3489   /S/SubmitForm%
3490   /F<<%
3491   /FS/URL%
3492   /F(\Hy@escapestring{\Form@action})%
3493   >>%
3494   \Fld@submitflags
3495   >>%
3496   \Fld@additionalactions
3497 }
3498 \ExplSyntaxOn
3499 \def\PDFForm@Reset{%
3500   /Subtype/Widget%
3501   \Fld@annotflags
3502   \Fld@pageobjref
3503   \Fld@annotnames
3504   /FT/Btn%
3505   \Fld@flags

```

```

3506      /H/P%
3507      /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3508      \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3509          \ifx\Fld@bordercolor\relax\else 1\fi
3510          \space
3511      \else
3512          /MK<<%
3513          \ifnum\Fld@rotation=\z@
3514          \else
3515              /R~\Fld@rotation
3516          \fi
3517          \ifx\Fld@bordercolor\relax
3518          \else
3519              /BC[\Fld@bordercolor]%
3520          \fi
3521          >>%
3522      \fi
3523      /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3524      /A<</S/ResetForm>>%
3525      \Fld@additionalactions
3526      }%
3527
3528
3529      %these patterns are used in hyperref checks.
3530      %it is unclear if they are really useful and if a backend support is
3531      %needed.
3532      \str_case:VnF \c_sys_backend_str
3533      {
3534          { pdfmode }
3535          {
3536              \def\HyPat@ObjRef
3537              {
3538                  [0-9]*[1-9][0-9]*~0~R
3539              }
3540          }
3541          { dvipdfmx }
3542          {
3543              \def\HyPat@ObjRef
3544              {
3545                  @[\~]+
3546              }
3547          }
3548          { xdvipdfmx }
3549          {
3550              \def\HyPat@ObjRef
3551              {
3552                  @[\~]+
3553              }
3554          }
3555      }
3556      { %also set in hyperref sty, so probably not needed.
3557          \def\HyPat@ObjRef/{.+}
3558      }
3559

```

```

3560
3561 \ExplSyntaxOff
3562 % UF: removed Hy@writebookmark
3563 % \Hy@currentbookmarklevel{0}
3564 % \Hy@numberline
3565 % \__hypwritetorep
3566 % counter{bookmark@seq@number}
3567 % removed \HyPsd@SanitizeForOutFile, not needed
3568 % removed \currentpdfbookmark, defined by bookmark,
3569 % should use \newcommand there
3570 % removed \subpdfbookmark, defined by bookmark,
3571 % should use \newcommand there
3572 % removed \belowpdfbookmark, defined by bookmark,
3573 % should use \newcommand there
3574 % removed \pdfbookmark, defined by bookmark,
3575 % \BOOKMARK
3576 % \@BOOKMARK
3577 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3578 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3579 %% removed \ReadBookmarks / unneeded with bookmark.
3580 %% removed \Hy@OutlineName
3581 %% removed \check@bm@number
3582 %% removed \calc@bm@number
3583
3584 \ifHy@implicit
3585 \else
3586 \expandafter\endinput
3587 \fi
3588 \newlength\Hy@SectionHShift
3589 \def\Hy@SectionAnchorHref#1{%
3590 \ifx\protect\@typeset@protect
3591 \Hy__hypSectionAnchor{#1}%
3592 \fi
3593 }
3594 \DeclareRobustCommand*\Hy__hypSectionAnchor}[1]{%
3595 \leavevmode
3596 \hbox to 0pt{%
3597 \kern-\Hy@SectionHShift
3598 \Hy@raisedlink{%
3599 \hyper@anchorstart{#1}\hyper@anchorend
3600 }%
3601 \hss
3602 }%
3603 }
3604 \@ifundefined{hyper@nopatch@sectioning}
3605 {
3606 \let\H@old@ssect\@ssect
3607 \def\@ssect#1#2#3#4#5{%
3608 \Hy@MakeCurrentHrefAuto{section*}%
3609 \setlength{\Hy@SectionHShift}{#1}%
3610 \begingroup
3611 \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3612 \toks\tw@\expandafter{%
3613 \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%

```

```

3614         #5%
3615     }%
3616     \edef\x{\endgroup
3617         \the\toks0{\the\toks\tw}%
3618     }\x
3619 }
3620 \let\H@old@schapter\@schapter
3621 \def\@schapter#1{%
3622     \begingroup
3623         \let\mkboth\@gobbletwo
3624         \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3625         \Hy@raisedlink{%
3626             \hyper@anchorstart{\@currentHref}\hyper@anchorend
3627         }%
3628     \endgroup
3629     \H@old@schapter{#1}%
3630 }
3631 \@ifundefined{chapter}{%
3632     \let\Hy@org@chapter\@chapter
3633     \def\@chapter{%
3634         \def\Hy@next{%
3635             \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3636             \Hy@raisedlink{%
3637                 \hyper@anchorstart{\@currentHref}\hyper@anchorend
3638             }%
3639         }%
3640         \ifnum\c@secnumdepth>\m@ne
3641             \@ifundefined{if@mainmatter}%
3642             \iftrue{\csname if@mainmatter\endcsname}%
3643             \let\Hy@next\relax
3644         \fi
3645         \fi
3646         \Hy@next
3647         \Hy@org@chapter
3648     }%
3649 }
3650 \let\H@old@part\@part
3651 \begingroup\expandafter\expandafter\expandafter\endgroup
3652 \expandafter\ifx\csname chapter\endcsname\relax
3653     \let\Hy@secnum@part\z@
3654 \else
3655     \let\Hy@secnum@part\m@ne
3656 \fi
3657 \def\@part{%
3658     \ifnum\Hy@secnum@part>\c@secnumdepth
3659         \phantomsection
3660     \fi
3661     \H@old@part
3662 }
3663 \let\H@old@spart\@spart
3664 \def\@spart#1{%
3665     \Hy@MakeCurrentHrefAuto{part*}%
3666     \Hy@raisedlink{%
3667         \hyper@anchorstart{\@currentHref}\hyper@anchorend

```

```

3668 }%
3669 \H@old@spart{#1}%
3670 }
3671 \let\H@old@sect\@sect
3672 \def\@sect#1#2#3#4#5#6[#7]#8{%
3673   \ifnum #2>\c@secnumdepth
3674     \expandafter\@firstoftwo
3675   \else
3676     \expandafter\@secondoftwo
3677   \fi
3678   {%
3679     \Hy@MakeCurrentHrefAuto{section*}%
3680     \setlength{\Hy@SectionHShift}{#3}%
3681     \begingroup
3682     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3683     \toks\tw@\expandafter{%
3684       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3685       #8%
3686     }%
3687     \edef\x{\endgroup
3688       \the\toks@{\the\toks\tw@}%
3689     }\x
3690   }{%
3691     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3692   }%
3693 }
3694 {}{}
3695 \expandafter\def\csname Parent-4\endcsname{}
3696 \expandafter\def\csname Parent-3\endcsname{}
3697 \expandafter\def\csname Parent-2\endcsname{}
3698 \expandafter\def\csname Parent-1\endcsname{}
3699 \expandafter\def\csname Parent0\endcsname{}
3700 \expandafter\def\csname Parent1\endcsname{}
3701 \expandafter\def\csname Parent2\endcsname{}
3702 \expandafter\def\csname Parent3\endcsname{}
3703 \expandafter\def\csname Parent4\endcsname{}
3704 %%
3705 </package>
3706 <*colorscheme>
3707 % collected from https://tex.stackexchange.com/questions/525261/better-default-
3708 % colors-for-hyperref-links
3709 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3710 % linkcolor=
3711 %,filecolor=
3712 %,urlcolor=
3713 %,menucolor=
3714 %,runcolor=
3715 %,linkbordercolor=
3716 %,filebordercolor=
3717 %,urlbordercolor=
3718 %,menubordercolor=
3719 %,runbordercolor=
3720 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }

```

```

3721 {
3722     linkcolor      = [rgb]{1,0,0}, %red
3723     filecolor      = [rgb]{0,1,1}, %cyan
3724     urlcolor       = [rgb]{1,0,1}, %magenta
3725     menucolor      = [rgb]{1, 0, 0}, %red
3726     runcolor       = [rgb]{0,1,1}, %cyan
3727 %-----
3728     linkbordercolor = [rgb]{1, 0 ,0 },
3729     filebordercolor = [rgb]{0, .5, .5},
3730     urlbordercolor  = [rgb]{0, 1, 1},
3731     menubordercolor = [rgb]{1, 0, 0},
3732     runbordercolor  = [rgb]{0, .7, .7}
3733 }
3734
3735 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3736 {
3737     linkcolor      = [rgb]{0,0.2,0.6},
3738     filecolor      = [rgb]{0.8,0,0.8},
3739     urlcolor       = [rgb]{0.8,0,0.8},
3740     menucolor      = [rgb]{0,0.2,0.6},
3741     runcolor       = [rgb]{0.8,0,0.8},
3742 %----- %-----
3743     linkbordercolor = [rgb]{0,0.2,0.6},
3744     filebordercolor = [rgb]{0.8,0,0.8},
3745     urlbordercolor  = [rgb]{0.8,0,0.8},
3746     menubordercolor = [rgb]{0,0.2,0.6},
3747     runbordercolor  = [rgb]{0.8,0,0.8}
3748 }
3749
3750 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3751 { %two colors: intern/extern
3752     linkcolor      = [rgb]{0.79216, 0, 0.12549},
3753     filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3754     urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3755     menucolor      = [rgb]{0.79216, 0, 0.12549 },
3756     runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3757 %----- %-----
3758     linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3759     filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3760     urlbordercolor  = [rgb]{0.01961, 0.44314, 0.6902},
3761     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3762     runbordercolor  = [rgb]{0.01961, 0.44314, 0.6902 }
3763 }
3764
3765 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3766 { %all darkgray
3767     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3768     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3769     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3770     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3771     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3772 %----- %-----
3773     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3774     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },

```

```

3775     urlbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3776     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3777     runbordercolor = [rgb]{0.4 ,0.4 ,0.4 }
3778 }
3779
3780 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3781 { %dvipsnam.def
3782     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3783     filecolor      = [rgb]{1, 0, 0}, %Red
3784     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3785     menucolor      = [rgb]{1, 0, 0}, %Red
3786     runcolor       = [rgb]{1, 0, 0}, %Red
3787     %----- %-----
3788     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3789     filebordercolor = [rgb]{1, 0, 0}, %Red
3790     urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3791     menubordercolor = [rgb]{1, 0, 0}, %Red
3792     runbordercolor = [rgb]{1, 0, 0} %Red
3793 }
3794
3795 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3796 { %dvipsnam.def
3797     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3798     filecolor      = [rgb]{0, 1, 0}, %Green
3799     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3800     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3801     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3802     %----- %-----
3803     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3804     filebordercolor = [rgb]{0, 1, 0}, %Green
3805     urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3806     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3807     runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3808 }
3809
3810
3811 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3812 {
3813     linkcolor      = [rgb]{0.50196, 0, 0.02353},
3814     filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3815     urlcolor       = [rgb]{0.54118, 0, 0.52941},
3816     menucolor      = [rgb]{0.44706, 0.45882, 0},
3817     runcolor       = [rgb]{0.07451, 0.46667, 0.46275},
3818     %----- %-----
3819     linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3820     filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3821     urlbordercolor = [rgb]{0.724708, 0.4, 0.717646},
3822     menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3823     runbordercolor = [rgb]{0.444706, 0.680002, 0.67765}
3824 }
3825
3826 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3827 {
3828     linkcolor      = [rgb]{0,0,0},

```



```

3829     filecolor      = [rgb]{0,0,0},
3830     urlcolor       = [rgb]{0,0,0},
3831     menucolor      = [rgb]{0,0,0},
3832     runcolor       = [rgb]{0,0,0},
3833 %----- %-----
3834     linkbordercolor = [rgb]{0,0,0},
3835     filebordercolor = [rgb]{0,0,0},
3836     urlbordercolor  = [rgb]{0,0,0},
3837     menubordercolor = [rgb]{0,0,0},
3838     runbordercolor  = [rgb]{0,0,0}
3839 }
3840 </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	
\#	285, 809
\\$	284
\%	810
\-	2227
\.	571, 575, 577
@curropt commands:	
\@curropt:	2748, 2815
\[2227
\\	25, 26, 36, 37, 38, 46, 50, 60, 80, 87, 94, 101, 108, 123, 128, 129, 137, 138, 147, 148, 149, 150, 151, 158, 166, 286, 287, 960, 2750, 2817, 3157
_	569, 571, 575, 577
\]	2227
A	
\A	569, 2227
\Acrobatmenu	19, 179
\addcontentsline	13
\AddToDocumentProperties	394, 2222, 2325
\AddToHook	432, 449, 455
\AddToHookNext	204
\advance	2680, 2755, 2799, 2800
allcolors (hypersetup key)	1088
\author	2
B	
\b	573
\begingroup	215, 283, 381, 2586, 2715, 2744, 2756, 2790, 3179, 3610, 3622, 3651, 3681
\belowpdfbookmark	3572
\bgroup	280, 381
\BOOKMARK	3575
bookmarkstype (hypersetup key)	13
bool commands:	
\bool_if:NTF	273, 302, 361, 729, 753, 773, 793, 804, 840, 906, 971, 1049, 1059, 1312, 1323
\bool_if:nTF	921, 1997, 2022, 3316, 3424
\bool_lazy_and:nnTF	429
\bool_lazy_or:nnTF	1342, 1391
\bool_new:N	209, 210, 555, 559, 563
\bool_set_true:N	564
bordercolormodel (hypersetup key)	13, 1109
box commands:	
\box_dp:N	666, 2900, 2951, 2986
\box_ht:N	665, 2899, 2950, 2985
\box_new:N	469, 566
\box_set_dp:Nn	2194
\box_set_ht:Nn	2201
\box_use:N	1329, 2902, 2960, 2989
\box_use_drop:N	1334
\box_wd:N	664, 2898, 2949, 2984
\l_tmpa_box	2896, 2898, 2899, 2900, 2902, 2935, 2949, 2950, 2951, 2960, 2982, 2984, 2985, 2986, 2989
C	
\catcode	284, 285
\char	2834, 3031, 3043
\chardef	164
\cite	34
clist commands:	
\clist_item:nn	2228, 2257
\clist_map_function:nN	130, 139
\clist_map_inline:nn	2389
\clist_map_variable:nNn	2747, 2814
color commands:	
\color_export:nnN	41, 440, 1008, 1130
\color_select:n	1016, 1052, 1333, 2941, 2945
\color_select:nn	1022
\color_set:nn	2, 399, 1033
\color_set:nnn	2, 398, 1039
color names:	
hyp/annot/file	552
hyp/annot/link	552
hyp/annot/menu	552
hyp/annot/run	552
hyp/annot/url	552
colorfile (hypersetup key)	1088
colorlink (hypersetup key)	1088
colorlinks (hypersetup key)	1065
colormenu (hypersetup key)	1088
colorrun (hypersetup key)	1088
colorscheme (hypersetup key)	1, 1500
colorurl (hypersetup key)	1088
cs commands:	
\cs_generate_variant:Nn	161, 162, 163, 396, 468, 607, 614, 837, 1008, 1025, 1042, 2228
\cs_gset:Npn	1304
\cs_gset_eq:NN	415, 426
\cs_if_exist:NTF	18

<code>\ExplSyntaxOn</code>	13, 16, 2618, 2675, 2699, 3059, 3231, 3252, 3305, 3374, 3412, 3498
extension (hypersetup key)	13, 1542
F	
<code>\fbox</code>	2853, 2941, 2945
<code>\fi</code>	12, 2574, 2601, 2604, 2613, 2713, 2721, 2722, 2753, 2771, 2774, 2775, 2781, 2784, 2793, 2795, 2802, 2805, 3132, 3154, 3164, 3166, 3172, 3176, 3183, 3184, 3193, 3197, 3201, 3205, 3209, 3213, 3217, 3221, 3225, 3229, 3246, 3250, 3349, 3350, 3351, 3358, 3362, 3366, 3368, 3370, 3389, 3393, 3397, 3401, 3408, 3473, 3474, 3481, 3485, 3487, 3508, 3509, 3516, 3520, 3522, 3587, 3592, 3644, 3645, 3656, 3660, 3677
file (hypersetup key)	10, 1573
file commands:	
<code>\file_input:n</code>	14
fileborderstyle (hypersetup key)	14, 1202
filecolor (hypersetup key)	1088
final (hypersetup key)	1524
<code>\fontencoding</code>	2830, 3027, 3039
<code>\fontfamily</code>	2829, 3026, 3038
<code>\fontseries</code>	2831, 3028, 3040
<code>\fontshape</code>	2832, 3029, 3041
fp commands:	
<code>\fp_eval:n</code>	2152, 2180, 2196
G	
<code>\gdef</code>	286, 287
<code>\GetDocumentProperties</code>	3
<code>\global</code>	2680, 3122
group commands:	
<code>\group_begin:</code>	271, 300, 323, 340, 359, 597, 735, 745, 759, 766, 798, 826, 845, 894, 911, 953, 976, 1002, 1051, 1315, 1332, 2458, 2828, 2878, 2921, 2969, 3002, 3025, 3037
<code>\group_end:</code>	314, 331, 348, 376, 604, 741, 745, 778, 782, 823, 826, 891, 894, 951, 953, 1000, 1002, 1061, 1335, 1338, 2465, 2835, 2914, 2962, 2995, 3032, 3044, 3054
H	
<code>\hbox</code>	381, 3596
hbox commands:	
<code>\hbox_overlap_right:n</code>	1329
<code>\hbox_set:Nn</code>	2896, 2935, 2982
<code>\hbox_set:Nw</code>	1316
<code>\hbox_set_end:</code>	1325
<code>\hbox_set_to_wd:Nnn</code>	2178
hidefile (hypersetup key)	1476
hidelink (hypersetup key)	1476
hidelinks (hypersetup key)	1476
hidemenu (hypersetup key)	1476
hiderun (hypersetup key)	1476
hideurl (hypersetup key)	1476
hook commands:	
<code>\hook_gput_code:nnn</code>	408, 419, 1045, 1055, 1308, 1319
<code>\hook_new:n</code>	397, 582, 678
<code>\hook_new_pair:nn</code>	266, 294, 318, 335, 353
<code>\hook_use:n</code>	270, 290, 299, 315, 322, 332, 339, 349, 358, 377, 599, 684, 693, 738, 762
<code>\href</code>	4, 5, 19, 268
<code>\hreflaunch</code>	5
<code>\hrefpdf</code>	5, 19, 319, 389
<code>\hrefrun</code>	5, 336, 390
<code>\hrefurl</code>	5, 19, 296, 388
<code>\hss</code>	3601
Hy internal commands:	
<code>\Hy_hyescapeform</code>	3143, 3156, 3162
<code>\Hy_hypSectionAnchor</code>	3591, 3594
HyField internal commands:	
<code>\HyField_hypAddToFields</code>	2580, 2607
hyp commands:	
<code>\l_hyp_annot_colorfile_bool</code> ..	7, 553
<code>\l_hyp_annot_colorlink_bool</code> ..	7, 553
<code>\l_hyp_annot_colormenu_bool</code> ..	7, 553
<code>\l_hyp_annot_colorrun_bool</code> ..	7, 553
<code>\l_hyp_annot_colorurl_bool</code> ..	7, 553
<code>\l_hyp_annot_ocgcolorfile_bool</code> ..	7, 557
<code>\l_hyp_annot_ocgcolorlink_bool</code> ..	7, 557
<code>\l_hyp_annot_ocgcolormenu_bool</code> ..	7, 557
<code>\l_hyp_annot_ocgcolorrun_bool</code> ..	7, 557
<code>\l_hyp_annot_ocgcolorurl_bool</code> ..	7, 557
<code>\l_hyp_current_dest_name_tl</code>	677, 736, 760
hyp internal commands:	
<code>\g__hyp_AcroForm_CoFields_prop</code> ..	2544, 2562, 2631, 2633, 2648
<code>\g__hyp_AcroForm_Fields_prop</code>	2545, 2555, 2626
<code>\l__hyp_annot_GoTo_bool</code>	729, 753, 773
<code>\l__hyp_annot_GoTo_bool\l__-</code> <code>hyp_annot_URI_bool\l__-</code> <code>hyp_annot_GoToR_bool\l__-</code> <code>hyp_annot_Named_bool\l__-</code> <code>hyp_annot_Launch_bool</code>	561
<code>\l__hyp_annot_GoToR_bool</code>	840
<code>\l__hyp_annot_Launch_bool</code>	906

\l__hyp_annot_Named_bool	971	\l__hyp_filename_tmpa_tl . . .	475 ,
\c__hyp_annot_types_seq			846 , 848 , 853 , 854 , 859 , 915 , 916 , 934
.	482 , 553 , 557 , 1088 , 1389 , 1487	__hyp_href_pdf_aux:nn	325 , 328
\l__hyp_annot_URI_bool	793	\l__hyp_href_pdf_destination_tl	
\g__hyp_bordercolormodel_str	212 , 223 , 262 , 330
.	442 , 552 , 1113 , 1132	\l__hyp_href_pdf_page_tl	213 , 231 , 871
__hyp_check_link_nesting:TF . . .		__hyp_href_run_aux:nn	342 , 345
.	638 , 646 ,	\l__hyp_href_run_parameter_tl . .	
	648 , 732 , 756 , 775 , 796 , 843 , 909 , 974	214 , 235 , 347
__hyp_citebordercolor_hook_-		__hyp_href_url_aux:n	368 , 371
init:	404 , 417 , 426	__hyp_href_url_aux:nn	309 , 311
__hyp_citecolor_hook_init:		\l__hyp_href_url_encode_bool . . .	
.	402 , 406 , 415	209 , 220 , 258 , 273 , 302 , 361
__hyp_clist_display:n . .	123 , 130 , 139	__hyp_href_url_format:	
__hyp_color_select:n	215 , 221 , 263 , 374
.	41 , 1009 , 1009 , 1025	\l__hyp_href_url_ismap_bool	
__hyp_color_select_aux:wn	210 , 234 , 804
.	1009 , 1013 , 1020	\l__hyp_href_url_protocol_tl . . .	
__hyp_color_set:nn . .	41 , 42 , 402 ,	211 , 222 , 261 , 313 , 375
	404 , 439 , 1026 , 1026 , 1042 , 1093 , 1129	__hyp_if_outer_link:	632
__hyp_color_set_aux:nwn		__hyp_if_outer_link:TF	648
.	1026 , 1030 , 1037	\l__hyp_link_Contents_tl	
\c__hyp_colorscheme_daleif_prop	3735	517 , 526 , 528 , 535 , 537 , 544 , 546
\c__hyp_colorscheme_henryford_-		__hyp_link_goto_begin:nw	
prop	3826	700 , 739 , 763
\c__hyp_colorscheme_julian_prop	3750	__hyp_link_goto_end: . .	722 , 740 , 777
\c__hyp_colorscheme_phelype_prop		\g__hyp_linknestlevel_int	
.	3811	631 , 634 , 731 , 747 , 755 , 784 ,
\c__hyp_colorscheme_szabolcsA_-			795 , 828 , 842 , 896 , 908 , 954 , 973 , 1003
prop	3780	g__hyp_linknestlevel_int	631
\c__hyp_colorscheme_szabolcsB_-		\c__hyp_map_annot_hyp_prop	482
prop	3795	\c__hyp_map_hyp_annot_prop	
\c__hyp_colorscheme_tivv_prop .	3765	482 , 1043 ,
\l__hyp_dest_box	33 ,		1116 , 1156 , 1183 , 1191 , 1202 , 1229 ,
	566 , 664 , 665 , 666 , 2178 , 2194 , 2201		1237 , 1306 , 1414 , 1450 , 1460 , 1573
\l__hyp_dest_name_tmpa_tl . .	475 ,	__hyp_ocg_init: 1248 , 1248 , 1304 , 1314	
	703 , 704 , 710 , 714 , 716 , 719 , 863 , 877	\l__hyp_optlang_regex	2226 , 2227 , 2259
\l__hyp_dest_pdfremotestartview_-		l__hyp_page/Trans	580
tl	501 , 872 , 1985 , 1989	__hyp_PageLabels_gpush: 615 , 615 , 628	
\g__hyp_dest_pdfstartpage_tl . . .		\l__hyp_para_tmpa_tl	478 , 920 , 923 , 934
. . .	501 , 1996 , 1998 , 2005 , 2023 , 2030	\l__hyp_para_tmpa_tl _{uuu} \l__hyp_-	
\g__hyp_dest_pdfstartview_tl . . .		text_tmpa_str _{uuu} \g__hyp_text_-	
	501 , 1998 , 2005 , 2016 , 2020 , 2023 , 2030	tmpa_str	475
\l__hyp_dest_pdfview_tl		__hyp_property_record:nn	
.	551 , 682 , 691 ,	461 , 462 , 468 , 2684
	2147 , 2150 , 2157 , 2160 , 2161 , 2162 ,	__hyp_secondoftwoewithopt:wnn . .	
	2163 , 2164 , 2165 , 2170 , 2174 , 2210	387 , 388 , 389 , 390
\c__hyp_dest_startview_regex . . .		__hyp_setup_info_date_key:nn . .	
.	567 , 1983 , 2014	2311 , 2344 , 2345
\c__hyp_dest_undefined_tl		__hyp_setup_info_key:nn . .	2229 ,
.	481 , 709 , 710		2304 , 2305 , 2306 , 2307 , 2309 , 2310
__hyp_destination:nn		__hyp_store_metadata:nn	
.	33 , 651 , 651 , 682 , 691	391 , 396 , 1592 , 2236 ,
			2279 , 2324 , 2339 , 2348 , 2364 , 2432

_hyp_text_cleanup:N ..	587, 587, 601
\l_hyp_text_enc_dest_print_tl ..	504, 862
\l_hyp_text_enc_dest_tl ..	33, 504, 658, 715
\l_hyp_text_enc_file_print_tl ..	504, 914
\l_hyp_text_enc_info_print_tl ..	504, 525, 534, 543, 612
\l_hyp_text_enc_para_print_tl ..	504, 919
\l_hyp_text_enc_uri_print_tl ..	275, 278, 304, 307, 363, 366, 504, 801, 1584
_hyp_text_pdfstring:nnN ..	523, 532, 541, 595, 595, 607, 609, 612, 656, 713, 799, 860, 912, 917, 1584
_hyp_text_pdfstring_info:nN ..	610, 610, 614, 2262, 2265, 2285, 2294
_hyp_text_purify:nN ..	583, 583, 600
_hyp_text_string_from_unicode:nN ..	591, 591, 602
\g_hyp_text_tmpa_str ..	480, 603, 605
\l_hyp_text_tmpa_str ..	479, 600, 601, 602, 603
\l_hyp_tmpa_box ..	469, 1316, 1329, 1334
\l_hyp_tmpa_int ..	469
\l_hyp_tmpa_seq ..	469, 1983, 1985, 2014, 2016, 2137, 2138, 2142, 2144, 2168, 2175, 2176, 2177, 2186, 2188, 2198, 2203, 2259, 2260, 2265, 2635, 2637, 2643
\l_hyp_tmpa_str ..	469, 2262, 2265, 2267, 2269, 2285, 2286, 2288, 2294, 2295, 2299
\l_hyp_tmpa_tl ..	469, 659, 663, 671, 1133, 1137, 1584, 1585, 1590, 1981, 1983, 2012, 2014, 2144, 2145, 2152, 2235, 2236, 2237, 2257, 2262, 2278, 2279, 2285
\l_hyp_tmpb_tl ..	469, 2257, 2259
\l_hyp_uri_tmpa_tl ..	475, 802, 803
hyp/anchor ..	677
hyp/annot/file (color name) ..	552
hyp/annot/link (color name) ..	552
hyp/annot/menu (color name) ..	552
hyp/annot/run (color name) ..	552
hyp/annot/url (color name) ..	552
hyp/text/pdfstring ..	582
\hypercalcbp ..	12, 18, 167
\HyperDestNameFilter ..	13, 657, 714
\hypersetup 1, 2, 5, 9, 10, 13, 19, 69, 102, 186	
\hypersetup keys:	
allcolors ..	1088
bookmarkstype ..	13
bordercolormodel ..	13, 1109
colorfile ..	1088
colorlink ..	1088
colorlinks ..	1065
colormenu ..	1088
colorrun ..	1088
colorscheme ..	1, 1500
colorurl ..	1088
debug ..	1524
destlabel ..	13
draft ..	1524
extension ..	13, 1542
file ..	10, 1573
fileborderstyle ..	14, 1202
filecolor ..	1088
final ..	1524
hidefile ..	1476
hidelink ..	1476
hidelinks ..	1476
hidemenu ..	1476
hiderun ..	1476
hideurl ..	1476
hypertextnames ..	1542
link ..	10, 1573
linkborder ..	14
linkborderstyle ..	14, 1202
linkcolor ..	1088
linkfileprefix ..	1542
linktoc ..	1542
linktocpage ..	1542
localanchorname ..	1542
menu ..	10, 1573
menuborder ..	14
menuborderstyle ..	14, 1202
menucolor ..	1088
naturalnames ..	1542
nested-links ..	10
nesting ..	14
ocgcolorfile ..	1342
ocgcolorlink ..	1342
ocgcolorlinks ..	1342
ocgcolormenu ..	1342
ocgcolorrun ..	1342
ocgcolorurl ..	1342
pageanchor ..	1542
pdfauthor ..	2226
pdfborder ..	14
pdfborderstyle ..	14, 1202
pdfcreationdate ..	14, 2311
pdfcreator ..	2226
pdfencoding ..	1514
pdfinfo ..	2378
pdfkeywords ..	2226

pdflang	14, 2215	\int_new:N	473, 631
pdflinkmargin	14	iow commands:	
pdfmetadate	14, 2311	\iow_newline:	3075, 3081, 3094
pdfmoddate	14, 2311		
pdfproducer	2226		
pdfremotestartview	12	K	
pdfstartview	12	\kern	3597
pdfsubject	2226	keys commands:	
pdftitle	2226	\keys_define:nn	192, 200, 218, 256,
pdftrapped	2350		400, 642, 1065, 1090, 1097, 1109,
pdfversion	1514		1118, 1143, 1158, 1177, 1204, 1223,
pdfview	12, 2133		1346, 1360, 1372, 1395, 1401, 1416,
plainpages	1542		1445, 1476, 1489, 1500, 1514, 1524,
run	10, 1573		1529, 1542, 1554, 1575, 1580, 2133,
runborder	14		2215, 2231, 2274, 2313, 2328, 2346,
runborderstyle	14, 1202		2350, 2378, 2430, 2436, 2449, 2469
runcolor	1088	\l_keys_key_str	197, 2299, 2534
unicode	1514	\keys_set:nn	189,
url	10, 1573		226, 245, 272, 301, 324, 341, 360,
urlborder	14		410, 421, 1506, 1513, 2355, 2382,
urlborderstyle	14, 1202		2385, 2386, 2387, 2388, 2459, 2534
urlcolor	1088	\keys_set_known:nn	2537
verbose	1524	\kvsetkeys	188, 2622
hypertextnames (hypersetup key)	1542		
hypListbox internal commands:		L	
__hypListbox	2803, 2862	\label	9, 13
hypRadio internal commands:		\LayoutCheckField	3015
__hypRadio	2788, 2810	\LayoutChoiceField	2785
hypwritetorep internal commands:		\LayoutPushButtonField	2891, 2908
__hypwritetorep	3565	\LayoutTextField	2723
		\leavevmode	2724, 2821, 3595
		legacy commands:	
I		\legacy_if:nTF	174, 447, 457,
\ifcase	3349, 3473, 3508		623, 2660, 2887, 2928, 2978, 3011, 3295
\ifcsname	3126	\let	388,
\ifdim	2753, 2792, 2793		389, 390, 749, 818, 830, 887, 898,
\ifnum	3180, 3349, 3355, 3386, 3473,		947, 956, 1005, 2547, 2548, 2575,
	3478, 3508, 3513, 3640, 3658, 3673		2671, 2672, 2673, 2705, 2706, 2740,
\iftrue	3642		2741, 2783, 3122, 3129, 3131, 3136,
\ifx	2584, 2722, 2782,		3153, 3187, 3606, 3620, 3623, 3632,
	3157, 3169, 3173, 3190, 3194, 3198,		3643, 3650, 3653, 3655, 3663, 3671
	3202, 3206, 3210, 3214, 3218, 3222,	link (hypersetup key)	10, 1573
	3226, 3243, 3247, 3350, 3351, 3359,	linkborder (hypersetup key)	14
	3363, 3370, 3390, 3394, 3401, 3402,	linkborderstyle (hypersetup key)	14, 1202
	3474, 3482, 3509, 3517, 3590, 3652	linkcolor (hypersetup key)	1088
\immediate	2568, 2571	linkfileprefix (hypersetup key)	1542
int commands:		linktoc (hypersetup key)	1542
\int_compare:nNnTF	634, 2142,	linktocpage (hypersetup key)	1542
	2168, 2854, 3272, 3325, 3435, 3457	localanchorname (hypersetup key)	1542
\int_compare_p:nNn	3318, 3426	\long	12
\int_eval:n	870		
\int_gdecr:N		M	
	747, 784, 828, 896, 954, 1003	\MakeButtonField	
\int_gincr:N	731, 755, 795, 842, 908, 973		2896, 2911, 2935, 2982, 2993
\int_max:nn	871	\MakeChoiceField	2871

\MakeFieldObject	2538	\paperwidth	3
\MakeRadioField	2853	\PassOptionsToPackage	459, 1534, 1539
\MakeTextField	2732	pdf commands:	
\mbox	1326	\pdf_bdcobject:nn	1328, 1331
menu (hypersetup key)	10, 1573	\pdf_destination:nn	27, 161, 670
menuborder (hypersetup key)	14	\pdf_destination:nnnn	663
menuborderstyle (hypersetup key)	14, 1202	\pdf_emc:	1330, 1336
menucolor (hypersetup key)	1088	\pdf_name_from_unicode_e:n	251, 850, 981
mode commands:		\pdf_object_if_exist:nTF	848
\mode_if_horizontal:TF	653, 675	\pdf_object_new:n	1250, 1251, 1252, 1253, 3062, 3063, 3064
\mode_leave_vertical:	269, 298, 321, 338, 357, 702, 812, 879, 936, 979, 2865, 2893, 2910, 2932, 2974, 3017	\pdf_object_ref:n	162, 859, 1256, 1258, 1280, 1281, 1284, 1288, 1293, 1298, 1303, 2653, 2655, 3120
msg commands:		\pdf_object_ref_last:	881, 2464
\msg_error:nn	431, 2907, 2992	\pdf_object_unnamed_write:nn	880, 2460
\msg_info:nnn	2243, 2248	\pdf_object_write:nnn	1254, 1260, 1270, 1282, 3065, 3107, 3114
\msg_line_context:	88	\pdf_pageobject_ref:n	163, 2005, 2030, 2692
\g_msg_module_name_prop	17	\pdf_string_from_unicode:nnN	593
\msg_new:nnn	56, 63, 68, 72, 76, 83, 90, 97, 104, 111, 117, 124, 133, 143, 154	\pdf_version:	1666, 1785, 1814, 1848, 1863, 1889, 1914, 1949, 1965, 2070, 2097
\msg_new:nnnn	20, 31, 42	\pdf_version_compare:NnTF	1656, 1747, 1769, 1805, 1838, 1853, 1879, 1904, 1932, 1955, 2060, 2087
\msg_warning:nn	176, 1520	\pdf_version_compare_p:Nn	925, 1343, 1392
\msg_warning:nnn	185, 706, 997	\pdf_version_major:	173, 1344, 1367, 1393, 1409
\msg_warning:nnnn	1364, 1405, 1618, 1662, 1696, 1719, 1781, 1810, 1844, 1859, 1872, 1885, 1910, 1945, 1961, 1988, 2019, 2052, 2066, 2093, 2127, 2209	\pdf_version_minor:	172, 1367, 1409
\msg_warning:nnnnn	196, 1436, 1470, 1562, 1638, 1675, 1762, 1795, 1820, 1896, 1923, 1974, 2079, 2106, 2372, 2476, 2495, 2514, 2525	pdfannot commands:	
N		\pdfannot_box:nnnn	2727, 2838, 2868, 2897, 2948, 2983, 3047
naturalnames (hypersetup key)	1542	\pdfannot_box_ref_last:	2582, 2598
nested-links (hypersetup key)	10	\pdfannot_dict_put:nnn	527, 536, 545, 813, 881, 937, 982, 1134, 1169, 1193, 1215, 1239, 1421, 1452
nesting (hypersetup key)	14	\pdfannot_dict_remove:nn	1124, 1164, 1185, 1210, 1231, 1429, 1462
\newcommand	170, 3569, 3571, 3573	\pdfannot_link:nnn	814, 882, 938, 983
\newcount	2676, 2861	\pdfannot_link_goto_begin:nw	719
\NewDocumentCommand	2538	\pdfannot_link_goto_end:	724
\NewExpandableDocumentCommand	387	\pdfannot_link_margin:n	8, 1700
\newlength	3588	\c_pdfannot_link_types_seq	561
\noexpand	2759, 2760, 2761, 2762, 3140	pdfauthor (hypersetup key)	2226
\nolinkurl	4	\pdfbookmark	3574
O		pdfborder (hypersetup key)	14
ocgcolorfile (hypersetup key)	1342	pdfborderstyle (hypersetup key)	14, 1202
ocgcolorlink (hypersetup key)	1342	pdfcreationdate (hypersetup key)	14, 2311
ocgcolorlinks (hypersetup key)	1342	pdfcreator (hypersetup key)	2226
ocgcolormenu (hypersetup key)	1342		
ocgcolorrun (hypersetup key)	1342		
ocgcolorurl (hypersetup key)	1342		
P			
pageanchor (hypersetup key)	1542		

\pdfdest	11	\pdfstringdefDisableCommands	388, 389, 390
pdfdict commands:		pdfsubject (hypersetup key)	2226
\pdfdict_new:n	580, 787, 833, 900, 965	pdftitle (hypersetup key)	2226
\pdfdict_put:nnn	243, 250, 581, 788, 789, 803, 806, 834, 835, 850, 856, 867, 877, 901, 902, 916, 931, 966, 967, 980, 1726, 1727, 1731, 1732, 2473, 2483, 2487, 2490, 2492, 2511, 2520, 2522	pdftrapped (hypersetup key)	2350
\pdfdict_remove:nn	240, 928, 1736, 1737	pdfversion (hypersetup key)	1514
\pdfdict_use:n	813, 880, 937, 943, 982, 987, 2462	pdfview (hypersetup key)	12, 2133
pdfencoding (hypersetup key)	1514	pdfxform commands:	
\pdfescapestring	3129	\pdfxform_if_exist:nTF	2824, 2936, 3020
pdffile commands:		\pdfxform_new:nnn	2540, 2826, 2939, 2943, 3022, 3034
\pdffile_embed_file:nnn	837, 851	\pdfxform_ref:n	2848, 2849, 2955, 2956, 3267, 3268
PDFForm internal commands:		\phantom	3043
\PDFForm_hypName	3168, 3171, 3175, 3178	\phantomsection	13, 19, 3659
pdfinfo (hypersetup key)	2378	plainpages (hypersetup key)	1542
pdfkeywords (hypersetup key)	2226	prg commands:	
pdflang (hypersetup key)	14, 2215	\prg_do_nothing:	415, 426
pdflinkmargin (hypersetup key)	14	\prg_generate_conditional_	
pdfmanagement commands:		variant:Nnn	2687
\pdfmanagement_add:nn	1709	\prg_new_conditional:Npnn	632
\pdfmanagement_add:nnn	617, 1280, 1281, 1303, 1590, 1610, 1626, 1630, 1650, 1658, 1687, 1742, 1752, 1776, 1790, 1802, 1807, 1832, 1840, 1855, 1881, 1906, 1940, 1957, 2003, 2028, 2042, 2062, 2089, 2118, 2221, 2244, 2249, 2269, 2288, 2298, 2322, 2337, 2359, 2445, 2464, 2628, 2645, 2652, 2654, 2656, 2662	\prg_return_false:	635
\pdfmanagement_if_active_p:	430	\prg_return_true:	636
\pdfmanagement_remove:nn	1587, 1606, 1614, 1634, 1646, 1671, 1683, 1691, 1705, 1714, 1758, 1773, 1792, 1829, 1868, 1893, 1919, 1936, 1970, 2000, 2025, 2038, 2047, 2075, 2102, 2114, 2123, 2253, 2282, 2319, 2334, 2442, 2455, 2666	prop commands:	
\pdfmanagement_show:n	2629	\prop_const_from_keyval:Nn	484, 492, 3720, 3735, 3750, 3765, 3780, 3795, 3811, 3826
pdfmeta commands:		\prop_gput:Nnn	17, 2555, 2562
\pdfmeta_standard_verify:nnTF	977, 2884, 2975	\prop_if_empty:Nn	2631
\pdfmeta_standard_verify:nTF	2658, 3236	\prop_item:Nn	2648
pdfmetadate (hypersetup key)	14, 2311	\prop_map_inline:Nn	1043, 1116, 1156, 1183, 1191, 1202, 1229, 1237, 1306, 1414, 1450, 1460, 1504, 1573, 2626, 2633
pdfmoddate (hypersetup key)	14, 2311	\prop_new:N	2544, 2545
pdfproducer (hypersetup key)	2226	property commands:	
pdfremotestartview (hypersetup key)	12	\property_if_recorded:nn	2687
pdfstartview (hypersetup key)	12	\property_if_recorded:nnTF	2690
\pdfstringdef	8, 9, 23, 29, 3185	\property_record:nn	465
		\property_ref:nn	2694
		\property_ref_undefined_warn:nn	2685
		\protect	749, 818, 830, 887, 898, 947, 956, 1005, 3590
		\providecommand	168, 169, 2569, 2572, 2737, 3125
		\ProvidesFile	3, 7
		R	
		\ReadBookmarks	3579
		\refstepcounter	12
		regex commands:	
		\regex_const:Nn	567
		\regex_extract_once:NnN	2259

<code>\regex_extract_once:NnNTF</code>	1983, 2014	<code>\str_case:nnTF</code>	2239, 3532
<code>\regex_new:N</code>	2226	<code>\str_case_e:nnTF</code>	2138
<code>\regex_set:Nn</code>	2227	<code>\str_compare:nNnTF</code>	2639
<code>\relax</code>	749, 818, 830, 887, 898, 947, 956, 1005, 2740, 2741, 2782, 3122, 3169, 3173, 3243, 3247, 3350, 3351, 3359, 3363, 3390, 3394, 3474, 3482, 3509, 3517, 3643, 3652	<code>\str_gset:Nn</code>	1113
<code>\RemoveFromHook</code>	1596	<code>\str_gset_eq:NN</code>	603
<code>\RenewDocumentCommand</code>	179	<code>\str_head:n</code>	2361, 2366
<code>\RequirePackage</code>	11, 39, 451, 3577	<code>\str_if_eq:nnTF</code>	660, 2267, 2286, 2295, 3234
<code>run</code> (hypersetup key)	10, 1573	<code>\str_if_eq_p:nn</code>	923, 1344, 1393
<code>runborder</code> (hypersetup key)	14	<code>\str_lowercase:n</code>	2138, 2279, 2339, 2362, 2367
<code>runborderstyle</code> (hypersetup key)	14, 1202	<code>\str_new:N</code>	474, 479, 480, 552
<code>runcolor</code> (hypersetup key)	1088	<code>\str_set:Nn</code>	585
S		<code>\str_set_convert:Nnnn</code>	8
<code>\selectfont</code>	2833, 3030, 3042	<code>\str_set_eq:NN</code>	605
seq commands:		<code>\str_tail:n</code>	2362, 2367
<code>\seq_const_from_clist:Nn</code>	482	<code>\str_uppercase:n</code>	2355, 2361, 2366
<code>\seq_count:N</code>	2142, 2168	<code>\string</code>	2569, 2572, 2593, 2611, 3145, 3149, 3160
<code>\seq_get_right:NN</code>	2144	<code>\subpdfbookmark</code>	3570
<code>\seq_if_empty:NTF</code>	2260	sys commands:	
<code>\seq_item:Nn</code>	1985, 2016, 2138, 2186, 2188, 2198, 2203, 2265	<code>\c_sys_backend_str</code>	3532
<code>\seq_map_inline:Nn</code>	553, 557, 561, 1088, 1389, 1487, 2643	T	
<code>\seq_new:N</code>	472	T _E X and L ^A T _E X 2 _ε commands:	
<code>\seq_put_right:Nn</code>	2175, 2176, 2177, 2635	<code>\@BOOKMARK</code>	3576
<code>\seq_set_split:Nnn</code>	2137	<code>\@CheckBox</code>	2998
<code>\seq_sort:Nn</code>	2637	<code>\@ChoiceMenu</code>	2738
<code>\setbox</code>	381	<code>\@Form</code>	2620
<code>\setlength</code>	3609, 3680	<code>\@PushButton</code>	2876
<code>\setpdfinkmargin</code>	8	<code>\@Reset</code>	2965
<code>\settowidth</code>	2752	<code>\@Submit</code>	2917
<code>\show</code>	2812	<code>\@TextField</code>	2703
socket commands:		<code>\@bookmarksopenlevel</code>	1601
<code>\socket_assign_plug:nn</code>	548, 549, 550	<code>\@bsphack</code>	464
<code>\socket_new:nn</code>	518, 519, 520	<code>\@chapter</code>	3632, 3633
<code>\socket_new_plug:nnn</code>	521, 530, 539	<code>\@currDisplay</code>	2752, 2820
<code>\socket_use:nn</code>	808, 864	<code>\@currValue</code>	2818, 2848
<code>\socket_use:nnn</code>	737, 761	<code>\@currentHref</code>	3613, 3626, 3637, 3667, 3684
sort commands:		<code>\@curropt</code>	2737, 2747, 2750, 2814, 2817
<code>\sort_return_same:</code>	2641	<code>\@empty</code>	437, 2547, 2548, 2584, 2671, 2672, 2673, 2705, 2706, 2722, 3190, 3194, 3198, 3202, 3206, 3210, 3214, 3218, 3222, 3226, 3370, 3401, 3402
<code>\sort_return_swapped:</code>	2640	<code>\@endForm</code>	2671
<code>\space</code>	2820, 3352, 3369, 3370, 3400, 3401, 3475, 3507, 3510	<code>\@esphack</code>	466
<code>\spacefactor</code>	653, 675	<code>\@firstofone</code>	2575, 3131, 3136, 3153
<code>\stockwidth</code>	3	<code>\@firstoftwo</code>	3674
str commands:		<code>\@for</code>	2748, 2815
<code>\c_colon_str</code>	960	<code>\@gobbles</code>	3623
<code>\c_hash_str</code>	809	<code>\@hyper@launch</code>	40, 960
<code>\c_percent_str</code>	810	<code>\@ifnextchar</code>	280
		<code>\@ifundefined</code>	3145, 3604, 3631, 3641

\@mainaux	2568, 2571, 2591, 2610	\Fld@flags	3260, 3313, 3346, 3382, 3421, 3470, 3505
\@mkboth	3623	\Fld@format@code	3194, 3196
\@ne	2680	\Fld@height	2708, 2729, 2732, 2743, 2801, 2840, 2853, 2869, 2871, 2920, 2968, 3004, 3049
\@nil	3143, 3156, 3162	\Fld@keystroke@code	3190, 3192
\@part	3650, 3657	\Fld@listcount	2811, 2819, 2854, 2861
\@pdfauthor	23	\Fld@mappingname	3173, 3175, 3247, 3249
\@pdfborder	168	\Fld@maxlen	3457, 3459
\@pdfborderstyle	169	\Fld@menulength	2745, 2751, 2798
\@savsf	653, 675	\Fld@name	2704, 2739, 2877, 3000, 3168, 3242
\@schapter	3620, 3621	\Fld@onblur@code	3210, 3212
\@secondoftwo	3676	\Fld@onclick@code	3335
\@sect	3671, 3672	\Fld@onenter@code	3222, 3224
\@spart	3663, 3664	\Fld@onexit@code	3226, 3228
\@ssect	3606, 3607	\Fld@onfocus@code	3206, 3208
\@tempdima	2746, 2753, 2755, 2792, 2793, 2794, 2798, 2799, 2800, 2801	\Fld@onmousedown@code	3214, 3216
\@tempdimb	2752, 2753	\Fld@onmouseup@code	3218, 3220
\@typeset@protect	3590	\Fld@pageobjref	2673, 2688, 3125, 3257, 3310, 3343, 3379, 3418, 3467, 3502
\BKM@color	437, 443	\Fld@radiosymbol	3398
\c@secnumdepth	3640, 3658, 3673	\Fld@rotation	3272, 3274, 3318, 3325, 3327, 3349, 3355, 3357, 3386, 3388, 3426, 3435, 3437, 3473, 3478, 3480, 3508, 3513, 3515
\calc@bm@number	3582	\Fld@submitflags	3494
\check@bm@number	3581	\Fld@validate@code	3198, 3200
\define@key	434	\Fld@value	2706, 2722, 2741, 2782, 2783, 3455
\Fld@additionalactions	3232, 3302, 3336, 3372, 3409, 3456, 3496, 3525	\Fld@width	2707, 2721, 2728, 2732, 2742, 2781, 2792, 2794, 2839, 2853, 2869, 2871, 2889, 2919, 2930, 2967, 2979, 3003, 3013, 3048
\Fld@align	3261, 3347, 3422	\Fld@X@additionalactions	3189, 3234, 3237
\Fld@altname	3169, 3171, 3243, 3245	\Form@action	3492
\Fld@annotflags	3256, 3309, 3342, 3378, 3417, 3466, 3501	\H@old@part	3650, 3661
\Fld@annotnames	3241, 3258, 3311, 3344, 3380, 3419, 3468, 3503	\H@old@schapter	3620, 3629
\Fld@bcolor	3280, 3282, 3363, 3365, 3394, 3396, 3430, 3443, 3445	\H@old@sect	3671, 3682, 3691
\fld@bcolor	3351	\H@old@spart	3663, 3669
\Fld@bordercolor	3276, 3278, 3320, 3329, 3331, 3350, 3359, 3361, 3390, 3392, 3428, 3439, 3441, 3474, 3482, 3484, 3509, 3517, 3519	\H@old@ssect	3606, 3611
\Fld@borderstyle	3262, 3315, 3348, 3384, 3423, 3472, 3523	\href@	280, 286
\Fld@borderwidth	2799, 2800, 3262, 3315, 3348, 3384, 3423, 3472, 3523	\href@split	286, 287
\Fld@calculate@code	2584, 3202, 3204	\Hy@abspage	627
\Fld@calculate@sortkey	2595	\Hy@activeanchorfalse	698
\Fld@cbsymbol	3284	\Hy@activeanchortrue	689
\Fld@charsize	2798, 3288, 3369, 3400, 3451, 3507	\Hy@AtBeginDocument	2549, 2565
\Fld@checkequals	2750, 2817	\Hy@bookmarkstyle	1602
\Fld@choices	3371	\Hy@chapapp	3624, 3635
\Fld@color	3289, 3291, 3370, 3401, 3452	\Hy@colorlink	35
\Fld@default	2705, 2722, 2740, 2783, 2812, 2813, 3001, 3402, 3406, 3407, 3454	\Hy@currentbookmarklevel	3563

\Hy@DisableOption	178	\HyAnn@Count	2676, 2677, 2680, 2681, 2682, 2683, 2684, 2685, 2690, 2694
\Hy@drafttrue	1533	\HyField@AddToFields	2578, 2733, 2855, 2872, 2903, 2959, 2988, 3052
\Hy@escapeform	2726, 2823, 2867, 2895, 2934, 2981, 3019, 3127, 3134, 3137	\HyField@afields	2547
\Hy@escapestring	3129, 3131, 3136, 3139, 3140, 3143, 3144, 3153, 3156, 3192, 3196, 3200, 3204, 3208, 3212, 3216, 3220, 3224, 3228, 3284, 3335, 3398, 3454, 3455, 3492	\HyField@AfterAuxOpen	2549, 2575, 2608
\Hy@finaltrue	1538	\HyField@AuxAddToCoFields	2560, 2572, 2593
\Hy@FormObjects	2625, 3060, 3122	\HyField@AuxAddToFields	2553, 2569, 2611
\Hy@gtemp	3185, 3187	\HyField@cofields	2548
\Hy@href	280	\HyField@FlagsCheckBox	3010
\Hy@href@nextactionraw	244	\HyField@FlagsChoice	2791
\Hy@href@page	232	\HyField@FlagsPushButton	2886, 2926, 2977
\Hy@linkfileprefix	1547	\HyField@FlagsRadioButton	2787
\Hy@linktoc	1558	\HyField@FlagsSubmit	2927
\Hy@MakeCurrentHref	205	\HyField@FlagsText	2720
\Hy@MakeCurrentHrefAuto	3608, 3624, 3635, 3665, 3679	\HyField@PDFChoices	2864
\Hy@next	3634, 3643, 3646	\HyField@SetKeys	2716, 2757, 2760, 2779, 2879, 2922, 2970, 3005
\Hy@numberline	171, 3564	\HyPat@ObjRef	3536, 3543, 3550, 3557
\Hy@org@chapter	3632, 3647	\hyper@@link	288
\Hy@OutlineName	3580	\hyper@anchor	677
\Hy@OutlineRerunCheck	3578	\hyper@anchorend	677, 3599, 3626, 3637, 3667
\Hy@pdfmajorversion	173	\hyper@anchorstart	677, 3599, 3626, 3637, 3667
\Hy@pdfminorversion	172	\hyper@link	35, 727
\Hy@pdfstringtrue	30, 598	\hyper@linkend	35, 771
\Hy@pdfversion	3180	\hyper@linkfile	330, 838
\Hy@pstringdef	608, 3160	\hyper@linklaunch	39, 347, 904, 963
\Hy@PutCatalog	615	\hyper@linknamed	40, 181, 969
\Hy@raisedlink	3598, 3625, 3636, 3666	\hyper@linkstart	35, 751
\Hy@RestoreLastskip	674	\hyper@linkurl	313, 374, 791
\Hy@ReturnAfterFi	12, 3161	\hyper@normalise	280, 309, 325, 342, 368, 381
\Hy@safe@activestru	811, 2587	\HyPL@Labels	617, 627
\Hy@SaveLastskip	654	\HyPL@storePageLabel	615
\Hy@secnum@part	3653, 3655, 3658	\HyPsd@SanitizeForOutFile	3567
\Hy@SectionAnchorHref	3589, 3613, 3684	\if@files	2567, 2590, 2609
\Hy@SectionHShift	3588, 3597, 3609, 3680	\ifFld@combo	2766, 2796
\Hy@StepCount	2751, 2819	\ifFld@hidden	2721, 2781
\Hy@temp	2588, 2603, 3142, 3143	\ifFld@multiline	2709
\Hy@unicodedefalse	3182	\ifFld@popdown	2767
\Hy@VerboseAnchor	655	\ifFld@radio	2763, 2786
\Hy@VerboseLinkStart	734, 758	\ifHy@implicit	3584
\Hy@VerboseLinkStop	742, 779, 821, 889, 949, 993	\ifHy@pdfescapeform	3128, 3138
\Hy@VersionChecked	164	\ifHy@unicode	3181
\Hy@WrapperDef	651	\kv@set@family@handler	184
\Hy@xspace@end	739, 820, 888, 948, 992	\m@ne	3640, 3655
\HyAnn@AbsPageLabel	2672, 2678, 2725, 2822, 2866, 2894, 2933, 2980, 3018	\OBJ@OCG@view	170
		\p@	2755
		\pdf@ifdraftmode	2623

\PDF@SetupDoc	165	\tl_if_empty_p:N	1998, 2023
\PDFForm@Check	3019, 3051, 3253	\tl_if_eq:NnTF	2145
\PDFForm@List	2867, 2870, 3340	\tl_if_exist:NnTF ...	3329, 3439, 3443
\PDFForm@Name		\tl_if_exist_p:N ...	3320, 3428, 3430
.....	2719, 2780, 2883, 3009, 3167	\tl_if_head_eq_charcode:nNTF ...	
\PDFForm@Push	2895, 2901, 3306	1011, 1028
\PDFForm@Radio	2823, 2843, 3375	\tl_new:N	
\PDFForm@Reset	2981, 2987, 3499	. 211, 212, 213, 214, 470, 471, 475,	
\PDFForm@Submit	2934, 2953, 3464	476, 477, 478, 501, 502, 503, 504,	
\PDFForm@Text	2726, 2731, 3414	505, 506, 507, 508, 509, 517, 551, 677	
\protected@edef	703	\tl_set:Nn	231,
\strip@pt 3288, 3369, 3400, 3451, 3507		232, 244, 275, 278, 304, 307, 363,	
\toks@	3611, 3617, 3682, 3688	366, 393, 511, 512, 513, 514, 515,	
\tw@	3612, 3617, 3683, 3688	516, 683, 692, 736, 760, 846, 1981,	
\url@	381, 384	1985, 1989, 2012, 2147, 2150, 2157,	
\Url@def	382	2160, 2161, 2162, 2163, 2164, 2165,	
\Url@HyperHook	382	2170, 2174, 2210, 2235, 2257, 2278	
\XR@ext	1544	\tl_set_eq:NN	710
\z@ 381, 2677, 2746, 3349, 3355,		\tl_to_str:N	848, 854, 859
3386, 3473, 3478, 3508, 3513, 3653		\tl_to_str:n	26, 27, 37, 38, 51
\zref@labelbylist	2682	token commands:	
\zref@labelbyprops	2681	\token_to_str:N	47
\zref@refused	2683	\toks	3612, 3617, 3683, 3688
\texorpdfstring	30		
text commands:			
\text_expand:n ... 66, 846, 2235, 2278			
\text_purify:n	8, 585		
\textsf	2941, 2945		
\the	627,		
2681, 2682, 2683, 2684, 2685, 2690,			
2694, 2794, 2798, 2801, 3617, 3688			
\theH... ..	12		
\ThisShouldNotHappen	3147		
\title	2		
tl commands:			
\c_space_tl	171, 1257,		
1288, 1293, 1298, 2848, 2849, 2856,			
3071, 3073, 3083, 3085, 3087, 3090,			
3092, 3288, 3291, 3423, 3451, 3452			
\tl_const:Nn	481		
\tl_gput_right:Nn	627		
\tl_gset:Nn	1996, 2016, 2020		
\tl_if_blank:nTF			
436, 865, 2280, 2317, 2332, 2440, 2453			
\tl_if_empty:NnTF			
704, 1585, 2237, 3276, 3280, 3289, 3452			
\tl_if_empty:nTF			
..... 238, 1122, 1162, 1181,			
1208, 1227, 1771, 1827, 1934, 2219			

	U
unicode (hypersetup key)	1514
\Url	4, 5, 215
url (hypersetup key)	10, 1573
\url	4–6, 23, 355, 381
urlborder (hypersetup key)	14
urlborderstyle (hypersetup key)	14, 1202
urlcolor (hypersetup key)	1088
\urldef	5, 22, 381, 383
\Urlfont	6
\UrlSpecial	4
use commands:	
\use:n	958
\use_i:nn	640, 646

	V
verbose (hypersetup key)	1524

	W
\write	2568, 2571, 2591, 2610

	X
\x	2758, 2778, 3616, 3618, 3687, 3689