

The `hyperref-generic` module

A generic driver for `hyperref`

The L^AT_EX Project*

Version 0.96y, released 2026-01-23

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):

```
\RequirePackage{pdfmanagement} %loads the PDF management and activates it
\SetKeys[document/metadata]
{
  %% options
  %% e.g. pdf version,
  % pdfversion=1.7,
}
```

If a special backend is needed it can be passed as option:

```
\RequirePackage[backend=dvipdfmx]{pdfmanagement}
```

The `pdfmanagement` is also loaded if `\DocumentMetadata` is used e.g. for tagging

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

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

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

1 Avoiding transition problems

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

- Package options are processed at the end of the driver, Class options are ignored. But not every option already works as package options, in some cases `hyperref` interferes. So it is recommended for most options—with the exception of a few mentioned below in section 9—to set them in `\hypersetup`, not as package option.
- This driver uses the `l3color` module for the colors. All colors defined with `\color_set:mn` 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 comm
```

- 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 it can be passed with `\DocumentMetadata` or the `document/metadata` key:

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

or

```
\RequirePackage{pdfmanagement}  
\SetKeys[document/metadata]{lang=de-DE}
```

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

- The code supports two new keys now to set the reservation of rights relative to text & data mining (TDM): `pdftdmreservation` which should be either 0 or 1, and `pdftdmpolicy` which should be an url pointing to the policy, see <https://pdfa.org/resource/xmp-extension-schema-templates/#tdmrep>.

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 `pdfLATEX` 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 file 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
urlencode	<code>\hrefurl</code>	if set the code will convert the argument to percent encoding. This allows non-ascii input.
protocol	<code>\hrefurl</code> , <code>\url</code>	This sets a prefix/protocol that is added to the url, see below.
format	<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.
destination	<code>\href</code> , <code>\hrefpdf</code>	A destination name in the PDF
page	<code>\href</code> , <code>\hrefpdf</code>	destination page, default: 1
pdfremotestartview	<code>\href</code> , <code>\hrefpdf</code>	start view, default: Fit
ismap	<code>\href</code> , <code>\hrefurl</code>	see PDF reference
afrelationship	<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.
run-parameter	<code>\hreflaunch</code>	run parameter (see the PDF reference)
nextactionraw	various	puts a <code>/Next</code> entry in the action dictionary (see the PDF reference)

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

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

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

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

Some main changes are

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

7.1 Background information

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

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

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

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

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

While link colors have been traditionally more or less under the control of `hyperref`, the situation with other format options, like the font, is more complicated. The font

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

7.2 New Keys

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

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

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

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

The default is `phelype`.

7.3 Public interfaces

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

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

frenchlinks The option `frenchlinks` does nothing at all.

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

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

8 PDF strings

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

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

The new method is under heavy development!

Important differences here are

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

9 Package options from `hyperref`

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

Options that still must be set as package options are

- `backref`
- `CJKbookmarks` this key should not be used anymore. At some time it will be removed.
- `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
`url` (*setup key*) new driver offers also boolean keys `link`, `url`, `file`, `run` and `menu` which allow to locally
`file` (*setup key*) disable a link type. So e.g. `\hypersetup{link=false}\ref{abc}` will give a reference
`run` (*setup key*) without link (this is naturally also possible with `\ref*{abc}`). This disables also all
`menu` (*setup key*) 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` or with `\SetKeys[document/metadata`

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` or `\SetKeys[document/metadata]`:

```
\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
- `PushButtones`, which use the action `/S/JavaScript` are suppressed.
- `ResetButtons`, which use the action `/S/ResetForm` are suppressed.
- In widget annotations, the `/AA` dictionary is suppressed.

13 Destinations

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

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

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

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

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

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

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

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

- `Fit`, `FitB`, `FitH`, `FitV`, `FitBH`, `FitBV` which can be followed by a positive integer (separated by a space) or the keyword `null`. The number can be gives as a *dimension expression* surrounded with `\hypercalcbp`. The driver redefines this command to use `\dim_to_decimal_in_bp:n`.
 - `pdfview` will ignore the integer and any other arguments and calculate the expected coordinates as described above for `pdftex` with all supported engines and backends.
 - `pdfstartview` and `pdfremotestartview` will pass the optional number or keyword after expansion as absolute coordinate. Missing numbers will be filled up with `null`.
- `XYZ`. This can be followed (separated by spaces) by up to three positive integers or keywords `null` which are then taken as *top left zoom* in this order. *zoom* is a factor, so e.g. 0.5 will give a scaling of 50%.

- `pdfview` will use the last value as *zoom*, ignore all other values and calculate the expected coordinates as described above for `pdftex` with all supported engines and backends (this means it is possible to use `XYZ 2` to set a zoom of 200%, it is not necessary to fill in dummy values.)
- `pdfstartview` and `pdfremotestartview` will pass the optional numbers or keyword after expansion as absolute coordinates and zoom. Missing numbers will be filled up with `null`.

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

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

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

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

13.1 Names of destinations

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

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

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

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

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

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

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

14 Assorted key descriptions

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

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

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

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

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

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

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

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

will create

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

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

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

urlborderstyle (*setup key*)

fileborderstyle (*setup key*)

runborderstyle (*setup key*)

menuborderstyle (*setup key*)

	Key	Values	comment / example
	/Type	/Border	optional
	/W	<i><number></i>	Width of border line
	/S	/S	solid (default)
		/D	dash pattern can be set with /D
		/B	beveled
		/I	inset
		/U	underline
	/D	<i><array of numbers></i>	dash pattern (lines/gaps) (default [3])
<code>pdfcreationdate</code> (<i>setup key</i>)	Setting these keys is normally not needed. If they are used the values of the first		
<code>pdfmoddate</code> (<i>setup key</i>)	two keys are stored directly in the Info dictionary for <code>/Creationdate</code> and <code>/ModDate</code> .		
<code>pdfmetadate</code> (<i>setup key</i>)	All three keys are used in XMP-metadata. The values are converted to strings but not processed further, so they should have the correct PDF format without the enclosing parentheses, e.g. <code>D:20200202111111+01'00'</code> .		
<code>pdflinkmargin</code> (<i>setup key</i>)	As described in the <code>hyperref</code> manual the behaviour differs between the backends: with <code>dvips</code> it is possible to change links locally, <code>pdflatex</code> and <code>lualatex</code> work by page, with <code>dvipdfmx</code> the setting is global (and has to be done in the preamble).		
<code>pdflang</code> (<i>setup key</i>)	The key will work, but it is recommended to set the language in <code>\DocumentMetadata</code> or <code>\SetKeys[document/metadata]</code> instead.		

File I

hyperref-generic driver implementation

```

1 <@=hyp>
2 <*headertestphase>
3 \ProvidesFile{hgeneric-testphase.def}[2026-01-23 v0.96y %
4   generic Hyperref driver for the LaTeX PDF management bundle]
5 </headertestphase>
6 <*header>
7 \ProvidesFile{hgeneric.def}[2026-01-23 v0.96y %
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 \msg_new:nnnn
19 { hyp }
20 { missing-resource-management }
21 { The-PDF-resource-management-is-required-for-this-hyperref-driver! }
22 {
23   Activate-it-with \\
24   \tl_to_str:n{\DocumentMetadata{<options>}}\\
25   before~\tl_to_str:n{\documentclass} or \\
26   \RequirePackage{pdfmanagement}
27 }
```

The pdfversion should be set in \DocumentMetadata or \SetKeys[document/metadata]

```
28 \msg_new:nnnn
29 { hyp }
30 { pdfversion-disabled }
31 {
32   This-hyperref-driver-ignores-the-pdfversion-key!\\
33   Set-the-pdfversion-in~\token_to_str:N \DocumentMetadata\\or
34   with \tl_to_str:n{\SetKeys[document/metadata]}
35 }
36 {
37   For example:\\
38   \tl_to_str:n
39   {
40     \DocumentMetadata { pdfversion=1.7 }
41   }
42 }
```

A generic message for ignored keys.

```
43 \msg_new:nnn
44 { hyp }
45 { key-dropped }
46 {
47   This-hyperref-driver-ignores-the-key~#1!\\
48   Please-check-the-documentation.
49 }
```

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

```
50 \msg_new:nnn
51 { hyp }
52 { pdfa-no-push-button }
53 { PDF/A:~Push-button-with-JavaScript-is-prohibited }
54
55 \msg_new:nnn
56 { hyp }
57 { pdfa-no-reset-button }
58 { PDF/A:~Reset-action-is-prohibited }
```

pdf/A message for not allowed Named actions

```
59 \msg_new:nnn
60 { hyp }
61 { pdfa-no-named-action }
62 { PDF/A:~Named-action-#1-is-prohibited }
```


A message if the destination name is empty.

```
63 \msg_new:nnn
64 { hyp }
65 { empty-destination-name }
66 {
67   Empty-destination~name,\
68     using~'#1'
69 }
```

A message if the destination check fails

```
70 \msg_new:nnn
71 { hyp }
72 { invalid-destination-value }
73 {
74   Invalid~value~'#1'~of~'#2'  \
75   is~replaced~by~'Fit'~\msg_line_context:.
76 }
```

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

```
77 \msg_new:nnn
78 { hyp }
79 { ignore-deprecated-or-unknown-option-in-pdf-version }
80 {
81   Option~'#1'~is~unknown~or~deprecated~in\
82   pdf~version~#2.~Ignored.
83 }
84 \msg_new:nnn
85 { hyp }
86 { ignore-deprecated-or-unknown-value-in-pdf-version }
87 {
88   Value~'#1'~is~unknown~or~deprecated~in\
89   pdf~version~#2.~Ignored.
90 }
91 \msg_new:nnn
92 { hyp }
93 { replace-deprecated-or-unknown-value-in-pdf-version }
94 {
95   Value~'#1'~is~unknown~or~deprecated~in\
96   pdf~version~#2. Value~'#3'~is used instead.
97 }
```

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:

```
98 \msg_new:nnn
99 { hyp }
100 { unknown-key }
101 {
102   unknown~key~#2~of~module~'#1'~set~to~'#3'.
103 }
104 \msg_new:nnn
105 { hyp }
106 { unknown-key-to-Hyp }
107 {
108   ignored~in~family~Hyp~unknown~key~#1.
109 }
```

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

```

110 \cs_new:Npn \__hyp_clist_display:n #1 {*~#1\}
111 \msg_new:nnn
112 { hyp }
113 { unknown-choice }
114 {
115   Value-'#3'-is-invalid-for-key-'#1'.\}
116   The-key-accepts-only-the-choices\}
117   \clist_map_function:nN { #2 }\__hyp_clist_display:n
118 }
119
120 \msg_new:nnn
121 { hyp }
122 { unknown-choice+empty }
123 {
124   Value-'#3'-is-invalid-for-key-'#1'.\}
125   The-key-accepts-only-the-choices\}
126   \clist_map_function:nN { #2 }\__hyp_clist_display:n
127   An-empty-value-removes-the-setting.
128 }
129
130 \msg_new:nnn
131 { hyp }
132 { no-bool }
133 {
134   Value-'#2'-is-invalid-for-key-'#1'.\}
135   The-key-accepts-only-the-choices\}
136   *~true\}
137   *~false \}
138   *~and-an-empty-value-which-removes-the-setting.\}
139   No-value-is-equivalent-to-using-'true'.
140 }

```

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

```

141 \msg_new:nnn
142 { hyp }
143 { empty-info-value }
144 {
145   Empty-value-for-key-#1.\}
146   This-isn't-honored-by-all-backends.
147 }

```

2 Variants

```

148 \cs_generate_variant:Nn\pdf_destination:n {nf}
149 \cs_generate_variant:Nn\pdf_object_ref:n {e}
150 \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

```

151 \chardef\Hy@VersionChecked=1 %don't check the version!

```

```

152 %\cs_set_protected:Npn \PDF@SetupDoc{}
153 %\PDF@FinishDoc{}% dummy needed for hyperref ...

```

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

\hypercalcbp

```

154 \cs_set_eq:NN \hypercalcbp \dim_to_decimal_in_bp:n

```

(End of definition for \hypercalcbp. This function is documented on page 19.)

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

```

155 \providecommand\@pdfborder{}
156 \providecommand\@pdfborderstyle{}
157 \newcommand\OBJ@OCG@view {} % needed in hyperref
158 \def\Hy@numberline#1{#1\c_space_tl} %needed by bookmark

```

The pdfversion should be set in \DocumentMetadata or \SetKeys [document/metadata] but we must copy it to the hyperref command:

```

159 \cs_set_eq:NN \Hy@pdfminorversion \pdf_version_minor:
160 \cs_set_eq:NN \Hy@pdfmajorversion \pdf_version_major:
161 \legacy_if:nT { Hy@setpdfversion }
162 {
163   \msg_warning:nn { hyp }{ pdfversion-disabled }
164 }
165 \Hy@DisableOption{pdfversion}

```

\Acrobatmenu should use the new internal link command

```

166 \RenewDocumentCommand \Acrobatmenu { m m }
167 {
168   \hyper@linknamed {#1} {#2}
169 }

```

\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!

```

170 % TODO should go at some time ...
171 % \kv@set@family@handler{Hyp}
172 % { \msg_warning:nne {hyp}{unknown-key-to-Hyp}{#1} }
173 \cs_set_protected:Npn \hypersetup #1
174 {
175   %\kvsetkeys{Hyp} {#1}
176   \keys_set:nn { hyp }{ #1 }
177 }
178 % TODO for now unknown keys should only give warnings.
179 \keys_define:nn { hyp }
180 {
181   unknown .code:n =
182   {
183     \msg_warning:nneee { hyp } { unknown-key }
184     { hyp }{ \l_keys_key_str } { #1 }
185   }
186 }

```

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.

```

187 \keys_define:nn { hyp }
188 {
189   next-anchor .code:n =
190   {
191     \AddToHookNext{__hyp/dest/make}
192     {\Hy@MakeCurrentHref{#1}}
193   }
194 }

```

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`)

```

195
196 \bool_new:N \l__hyp_href_url_encode_bool
197 \bool_new:N \l__hyp_href_url_ismap_bool
198 \tl_new:N \l__hyp_href_url_protocol_tl
199 \tl_new:N \l__hyp_href_pdf_destination_tl
200 \tl_new:N \l__hyp_href_pdf_page_tl
201 \tl_new:N \l__hyp_href_run_parameter_tl
202 \cs_new_protected:Npn \__hyp_href_url_format: {\begingroup\Url}
203
204
205 \keys_define:nn { hyp / href }
206 {
207   ,urlencode .bool_set:N = \l__hyp_href_url_encode_bool
208   ,format .code:n = { \cs_set:Nn \__hyp_href_url_format: {#1} },
209   ,protocol .tl_set:N = \l__hyp_href_url_protocol_tl
210   ,destination .tl_set:N = \l__hyp_href_pdf_destination_tl
211   ,pdfremotestartview .code:n =
212   {
213     \keys_set:nn { hyp }
214     { pdfremotestartview = #1 }
215   }
216   ,page .code:n =
217   {
218     \tl_set:Nn \l__hyp_href_pdf_page_tl {#1}
219     \tl_set:Nn \Hy@href@page {#1}
220   }
221   ,ismap .bool_set:N = \l__hyp_href_url_ismap_bool
222   ,run-parameter .tl_set:N = \l__hyp_href_run_parameter_tl
223   ,nextactionraw .code:n =
224   { %perhaps some safety match later, see hyperref code
225     \tl_if_empty:nTF {#1}
226     {
227       \pdfdict_remove:nn{l_hyp/annot/A}{Next}
228     }
229     {
230       \pdfdict_put:nnn{l_hyp/annot/A}{Next}{#1}
231       \tl_set:Nn \Hy@href@nextactionraw {/Next~#1}
232       \keys_set:nn {hyp }{ pdfnewwindow = true}

```

```

233     }
234   }
235   ,afrelationship .code:n =
236   {
237     \pdfdict_put:nne
238     { l_pdffile/Filespec}{AFRelationship}{ \pdf_name_from_unicode_e:n {#1}}
239   }
240
241 }
242
243 \keys_define:nn { hyp }
244 {
245   ,href / urlencode .bool_set:N = \l__hyp_href_url_encode_bool
246   ,href / urlencode .default:n = {true}
247   ,href / urlencode .initial:n = {false}
248   ,href / protocol .tl_set:N = \l__hyp_href_url_protocol_tl
249   ,href / destination .tl_set:N = \l__hyp_href_pdf_destination_tl
250   ,href / format .code:n = { \cs_set:Nn \_hyp_href_url_format:{#1} }
251 }
252
253 \hook_new_pair:nn{cmd/href/before}{cmd/href/after}
254
255 \DeclareRobustCommand*{\href}[1][ ]{%
256   \mode_leave_vertical:
257   \hook_use:n{cmd/href/before}
258   \group_begin:
259   \keys_set:nn { hyp / href } {#1}
260   \bool_if:NTF \l__hyp_href_url_encode_bool
261   {
262     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
263   }
264   {
265     \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
266   }
267   \@ifnextchar\bgroup\Hy@href{\hyper@normalise\href@}%
268 }
269
270 \begingroup
271 \catcode`\$=6 %
272 \catcode`\#=12 %
273 \gdef\href@\$1{\expandafter\href@split$1##\}%
274 \gdef\href@split$1#$2#$3\\$4{%
275   \hyper@@link{$1}{$2}{$4}%<---_hyp-docstrip doubling!
276   \endgroup
277   \hook_use:n{cmd/href/after}
278 }%
279 \endgroup
280
281 \hook_new_pair:nn{cmd/hrefurl/before}{cmd/hrefurl/after}
282
283 \DeclareRobustCommand*{\hrefurl}[1][ ]
284 {
285   \mode_leave_vertical:
286   \hook_use:n{cmd/href/before}

```

```

287 \group_begin:
288 \keys_set:nn { hyp / href } {#1}
289 \bool_if:NTF \l__hyp_href_url_encode_bool
290 {
291   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
292 }
293 {
294   \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
295 }
296 \hyper@normalise\__hyp_href_url_aux:nn}
297
298 \cs_new_protected:Npn \__hyp_href_url_aux:nn #1 #2
299 {
300   \exp_args:Nno\hyper@linkurl{#2}{\l__hyp_href_url_protocol_tl#1}
301   \group_end:
302   \hook_use:n{cmd/href/after}
303 }
304
305 \hook_new_pair:nn{cmd/hrefpdf/before}{cmd/hrefpdf/after}
306 \DeclareRobustCommand*{\hrefpdf}[1] []
307 {
308   \mode_leave_vertical:
309   \hook_use:n{cmd/hrefpdf/before}
310   \group_begin:
311   \keys_set:nn { hyp / href } {#1}
312   \hyper@normalise\__hyp_href_pdf_aux:nn
313 }
314
315 \cs_new_protected:Npn \__hyp_href_pdf_aux:nn #1 #2
316 {
317   \exp_args:Nnno\hyper@linkfile{#2}{#1}{\l__hyp_href_pdf_destination_tl}
318   \group_end:
319   \hook_use:n{cmd/hrefpdf/after}
320 }
321
322 \hook_new_pair:nn{cmd/hrefrun/before}{cmd/hrefrun/after}
323 \DeclareRobustCommand*{\hrefrun}[1] []
324 {
325   \mode_leave_vertical:
326   \hook_use:n{cmd/hrefrun/before}
327   \group_begin:
328   \keys_set:nn { hyp / href } {#1}
329   \hyper@normalise\__hyp_href_run_aux:nn
330 }
331
332 \cs_new_protected:Npn \__hyp_href_run_aux:nn #1 #2
333 {
334   \exp_args:Nnno\hyper@linklaunch{#1}{#2}{\l__hyp_href_run_parameter_tl}
335   \group_end:
336   \hook_use:n{cmd/hrefrun/after}
337 }
338
339
340 \hook_new_pair:nn{cmd/url/before}{cmd/url/after}

```

```

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

```

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

```

367
368 \def\urldef#1#2{\begingroup\def\url{\hyper@normalise\url@}\setbox\z@\hbox\bgroup
369   \def\Url@HyperHook##1\endgroup{\Url@def{#1}{#2}}%
370   % Because hyperref breaks \urldef and does not define its own (Grrrr!)...
371   \def\url@##1{\egroup\endgroup\DeclareRobustCommand#1{#2{##1}}}%
372   #2}
373

```

make the new commands compatible with `\pdfstringdef`:

```

374 \NewExpandableDocumentCommand\__hyp_secondoftwowithopt:wnn {omm}{#3}
375 \pdfstringdefDisableCommands{\let\hrefurl\__hyp_secondoftwowithopt:wnn}
376 \pdfstringdefDisableCommands{\let\hrefpdf\__hyp_secondoftwowithopt:wnn}
377 \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.

```

378 \cs_new_protected:Npn \__hyp_store_metadata:nn #1 #2 %#1 key, #2 value.
379 {
380   %\tl_set:cn {#1}{#2}

```

```

381     \AddToDocumentProperties[hyperref]{#1}{#2}
382   }
383 \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.

```

384 \hook_new:n{hyp/link/cite}
385 %\color_set:nnn {hyp/color/cite}{HTML}{2E7E2A}
386 %\color_set:nn {hyp/color/citeborder}{hyp/color/cite!60!white}
387 \keys_define:nn { hyp }
388 {
389   ,citecolor .code:n = {\__hyp_color_set:ne {hyp/color/cite}{#1}\__hyp_citecolor_hook_init:}
390   ,citebordercolor
391     .code:n = {\__hyp_color_set:ne {hyp/color/citeborder}{#1}\__hyp_citebordercolor_hook_init:}
392 }
393 \cs_new_protected:Npn \__hyp_citecolor_hook_init:
394 {
395   \hook_gput_code:nnn { hyp/link/cite }{hyp/cite}
396   {
397     \keys_set:nn { hyp }
398     {
399       linkcolor      = hyp/color/cite
400     }
401   }
402 \cs_gset_eq:NN \__hyp_citecolor_hook_init: \prg_do_nothing:
403 }
404 \cs_new_protected:Npn \__hyp_citebordercolor_hook_init:
405 {
406   \hook_gput_code:nnn { hyp/link/cite }{hyp/citeborder}
407   {
408     \keys_set:nn { hyp }
409     {
410       linkbordercolor = hyp/color/citeborder
411     }
412   }
413 \cs_gset_eq:NN \__hyp_citebordercolor_hook_init: \prg_do_nothing:
414 }
415

```

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

```

416 \bool_lazy_and:nnF
417 { \cs_if_exist_p:N \pdfmanagement_if_active_p: }{ \pdfmanagement_if_active_p: }
418 { \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

```

419 \AddToHook { package/bookmark/after}
420 {
421   \define@key{BKM}{color}
422   {
423     \tl_if_blank:nTF {#1}
424     { \cs_set_eq:NN\BKM@color\@empty }
425     {
426       \__hyp_color_set:ne {__hyp/tmpa}{#1}
427       \color_export:nVN
428         {__hyp/tmpa}
429         \g__hyp_bordercolormodel_str
430         \BKM@color
431     }
432   }
433   \keys_define:nn {hyp}
434   {
435     bookmarksnumbered .legacy_if_set:n = {BKM@numbered}
436     ,bookmarksopen .code:n = {\bookmarksetup{open=#1}}
437     ,bookmarksopen .default:n = true
438     ,bookmarksdepth .code:n = {\bookmarksetup{depth=#1}}
439     ,bookmarksopenlevel .code:n = {\bookmarksetup{openlevel=#1}}
440   }
441 }
442 \legacy_if:nTF { Hy@bookmarks }
443 {
444   \AddToHook{begindocument/before}[hyperref/bookmark]
445   {
446     \RequirePackage{bookmark}
447   }
448 }

```

empty hook chunk to ensure that the chunk exists.

```

449 {
450   \AddToHook{begindocument/before}[hyperref/bookmark]{}
451 }
452 \legacy_if:nT {Hy@draft}
453 {
454   \PassOptionsToPackage{draft}{bookmark}
455 }

```

6 Reference and label commands

This uses the in-built property module.

```
\__hyp_property_record:nn
```

```

456 %
457 A label command which adds the space commands from LaTeX:
458 \cs_new_protected:Npn \__hyp_property_record:nn #1 #2 %label/attributes
459 {

```

```

459 \@bsphack
460 \property_record:nn{#1}{#2}
461 \@esphack
462 }

```

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.

```

463 \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

```

\_hyp_tmpa_tl
\_hyp_tmpb_tl
\_hyp_tmpa_seq
\_hyp_tmpa_int
\_hyp_tmpa_box
\_hyp_tmpa_str
464 \box_new:N \_hyp_tmpa_box
465 \tl_new:N \_hyp_tmpa_tl
466 \tl_new:N \_hyp_tmpb_tl
467 \seq_new:N \_hyp_tmpa_seq
468 \int_new:N \_hyp_tmpa_int
469 \str_new:N \_hyp_tmpa_str

```

(End of definition for _hyp_tmpa_tl and others.)

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

```

\_hyp_dest_name_tmpa_tl
\_hyp_uri_tmpa_tl
\_hyp_filename_tmpa_tl
\_hyp_text_tmpa_str \g_hyp_text_tmpa_str
TODO: document and check use!
470 \tl_new:N \_hyp_dest_name_tmpa_tl
471 \tl_new:N \_hyp_uri_tmpa_tl
472 \tl_new:N \_hyp_filename_tmpa_tl
473 \tl_new:N \_hyp_para_tmpa_tl
474 \str_new:N \_hyp_text_tmpa_str
475 \str_new:N \g_hyp_text_tmpa_str

```

(End of definition for _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.

```

476 \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 annot names to hyperref names and back.

```

\_hyp_map_annot_hyp_prop
\_hyp_map_hyp_annot_prop
477 \seq_const_from_clist:Nn \c_hyp_annot_types_seq
478 {url,link,file,menu,run}
479 \prop_const_from_keyval:Nn \c_hyp_map_annot_hyp_prop
480 {
481   URI = url,
482   GoTo = link,
483   GoToR = file,
484   Named = menu,

```

```

485     Launch= run
486   }
487 \prop_const_from_keyval:Nn \c__hyp_map_hyp_annot_prop
488   {
489     url    = URI,
490     link   = GoTo,
491     file   = GoToR,
492     menu   = Named,
493     run    = Launch
494   }
495

```

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

7.3 Variables

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

```

\g__hyp_dest_pdfstartview_tl
\l__hyp_dest_pdfremotestartview_tl

```

```

496 \tl_new:N \g__hyp_dest_pdfstartpage_tl
497 \tl_new:N \g__hyp_dest_pdfstartview_tl
498 \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
499 \tl_new:N \l__hyp_text_enc_uri_print_tl
\l__hyp_text_enc_dest_tl
500 \tl_new:N \l__hyp_text_enc_info_print_tl
\l__hyp_text_enc_dest_print_tl
501 \tl_new:N \l__hyp_text_enc_dest_tl
\l__hyp_text_enc_file_print_tl
502 \tl_new:N \l__hyp_text_enc_dest_print_tl
\l__hyp_text_enc_para_print_tl
503 \tl_new:N \l__hyp_text_enc_file_print_tl
504 \tl_new:N \l__hyp_text_enc_para_print_tl
505
506 \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
507 \tl_set:Nn \l__hyp_text_enc_info_print_tl {utf16/hex}
508 \tl_set:Nn \l__hyp_text_enc_dest_tl {utf8/string-raw}
509 \tl_set:Nn \l__hyp_text_enc_dest_print_tl {utf8/string}
510 \tl_set:Nn \l__hyp_text_enc_file_print_tl {utf8/string}
511 \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.

```

512 \tl_new:N\l__hyp_link_Contents_tl
513 \socket_new:nn {hyp/link/GoTo/Contents}{2}
514 \socket_new:nn {hyp/link/URI/Contents}{1}
515 \socket_new:nn {hyp/link/GoToR/Contents}{1}
516 \socket_new_plug:nnn {hyp/link/GoTo/Contents}{default}

```

```

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

```

546 \tl_new:N \l__hyp_dest_pdfview_tl

```

(End of definition for `\l__hyp_dest_pdfview_tl`.)

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

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

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

`\g__hyp_bordercolormodel_str` (*color name*)

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

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

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

```

547 \str_new:N \g__hyp_bordercolormodel_str

```

(End of definition for `\g__hyp_bordercolormodel_str`.)

7.4 Booleans

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

`\l_hyp_annot_colorurl_bool`

`\l_hyp_annot_colorfile_bool`

`\l_hyp_annot_colorryn_bool`

`\l_hyp_annot_colormenu_bool`

```

548 \seq_map_inline:Nn \c__hyp_annot_types_seq
549 {
550   \bool_new:c {l_hyp_annot_color#1_bool}
551 }

```

(End of definition for `\l_hyp_annot_colorlink_bool` and others. These variables are documented on page 8.)

`\l_hyp_annot_ocgcolorlink_bool` `\l_hyp_annot_ocgcolorurl_bool`
`\l_hyp_annot_ocgcolorfile_bool` `\l_hyp_annot_ocgcolorrrun_bool`
`\l_hyp_annot_ocgcolormenu_bool` These booleans are needed to control the ocgcolors. They are public so that other packages can query the state too.

```

552 \seq_map_inline:Nn \c__hyp_annot_types_seq
553 {
554   \bool_new:c {l_hyp_annot_ocgcolor#1_bool}
555 }

```

(End of definition for `\l_hyp_annot_ocgcolorlink_bool` and others. These variables are documented on page 8.)

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

```

556 \seq_map_inline:Nn \c_pdfannot_link_types_seq
557 {
558   \bool_new:c {l__hyp_annot_#1_bool}
559   \bool_set_true:c {l__hyp_annot_#1_bool}
560 }

```

(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.

```

561 \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.

```

562 \regex_const:Nn \c__hyp_dest_startview_regex
563 {
564   \A\ *
565   (?:
566     (?<XYZ (?<\ +(?:<:\d+|\d*\.\d+)|null))<3>\ )
567   |
568   (?<Fit\b|FitB\b)
569   |
570   (?<(?:FitH|FitV|FitBH|FitBV)<?<\ +(?:<:\d+|\d*\.\d+)|\ +null)<1>
571   |
572   (?<FitR (?<\ +\d+|\ +\d*\.\d+)<4>\ )
573   )
574 }

```

(End of definition for `\c__hyp_dest_startview_regex`.)

7.7 PDF dictionaries

`1__hyp_page/Trans` This dictionary is used for page transitions.

```
575 \pdfdict_new:n {1__hyp_page/Trans}
576 \pdfdict_put:nnm {1__hyp_page/Trans}{Type}{/Trans}
```

(End of definition for 1__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`

```
577 \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`

```
578 \cs_new_protected:Npn \__hyp_text_purify:nN #1 #2 %#1 input, #2 str command
579 {
580   \str_set:Ne #2 {\text_purify:n { #1 } }
581 }
```

(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`

```
582 \cs_new_protected:Npn \__hyp_text_cleanup:N #1
583 {
584
585 }
```

(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 `l3pdftools`.

```
#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`

```
586 \cs_new_protected:Npn \_hyp_text_string_from_unicode:nN #1 #2
587 {
588   \pdf_string_from_unicode:nVN { #1 } #2 #2
589 }
```

(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`

```
590 \cs_new_protected:Npn \_hyp_text_pdfstring:nnN #1 #2 #3
591 {
592   \group_begin:
593   \Hy@pdfstringtrue
594   \hook_use:n {hyp/text/pdfstring}
595   \_hyp_text_purify:nN { #1 } \l__hyp_text_tmpa_str
596   \_hyp_text_cleanup:N      \l__hyp_text_tmpa_str
597   \_hyp_text_string_from_unicode:nN { #2 } \l__hyp_text_tmpa_str
598   \str_gset_eq:NN \g__hyp_text_tmpa_str\l__hyp_text_tmpa_str
599   \group_end:
600   \str_set_eq:NN #3 \g__hyp_text_tmpa_str
601 }
602 \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

```
603 \cs_new_protected:Npn \Hy@pstringdef #1 #2
604 { \_hyp_text_pdfstring:enN {#2} {utf8/string-raw}#1 }
```

This is a special version for info keys:

`_hyp_text_pdfstring_info:nN`

```
605 \cs_new_protected:Npn \_hyp_text_pdfstring_info:nN #1 #2
606 {
607   \_hyp_text_pdfstring:noN { #1 }{\l__hyp_text_enc_info_print_tl } #2
608 }
609 \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: `hxtex.def`, line 80-110

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

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

dvipdfm: identical to xetex

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

luatex identical to pdftex

The code in `hyperref` inspects `\thepage` and tries to figure out the numbering system and the prefix. E.g. A-32 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
610 \cs_new_protected:Npn\__hyp_PageLabels_gpush:
611   {
612     \pdfmanagement_add:nne {Catalog} {PageLabels}{<</Nums [\HyPL@Labels]>>}
613   }
614
615 \def\Hy@PutCatalog #1 {}
616
617
618 \legacy_if:nT { Hy@pdfpagelabels }
619   {
620     \cs_set_protected:Npn \HyPL@StorePageLabel #1
621       {
622         \tl_gput_right:Nc \HyPL@Labels { \the\Hy@abspage<<#1>> }
623         \__hyp_PageLabels_gpush:
624       }
625   }
```

(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`

```

626 \int_new:N \g__hyp_linknestlevel_int
(End of definition for g__hyp_linknestlevel_int.)
627 \prg_new_conditional:Npnn \__hyp_if_outer_link: {TF}
628 {
629   \int_compare:nNnTF { \g__hyp_linknestlevel_int } > {1}
630   { \prg_return_false: }
631   { \prg_return_true: }
632 }
633 \cs_new:Npn \__hyp_check_link_nesting:TF #1 #2
634 {
635   \use_i:nn {#1}{#2}
636 }
637 \keys_define:nn { hyp }
638 {
639   nested-links .choice:,
640   nested-links / true .code:n =
641     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \use_i:nn },
642   nested-links / false .code:n =
643     { \cs_set_eq:NN \__hyp_check_link_nesting:TF \__hyp_if_outer_link:TF },
644   nested-links .default:n = {true}
645 }

```

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 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`

```

646 \Hy@WrapperDef \__hyp_destination:nn #1 #2
647 {
648   \mode_if_horizontal:T { \@savsf\spacefactor }
649   \Hy@SaveLastskip      %defined in hyperref
650   \Hy@VerboseAnchor{#1} %defined in hyperref, for debugging

```

```

651 \__hyp_text_pdfstring:eoN
652   { \HyperDestNameFilter{#1} }
653   { \l__hyp_text_enc_dest_tl }
654   \l__hyp_tmpa_tl
655 \str_if_eq:nnTF {#2} {fitrbox}
656   {
657     \exp_args:NV
658     \pdf_destination:nnnn \l__hyp_tmpa_tl
659     { \box_wd:N \l__hyp_dest_box }
660     { \box_ht:N \l__hyp_dest_box }
661     { \box_dp:N \l__hyp_dest_box }
662   }
663   {
664     \exp_args:NV
665     \pdf_destination:nf
666     { \l__hyp_tmpa_tl }
667     { #2 }
668   }
669 \Hy@RestoreLastskip %defined in hyperref
670 \mode_if_horizontal:T { \spacefactor\@savsf }
671 }

```

(End of definition for __hyp_destination:nn.)

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

```

\hyper@anchor
\hyper@anchorstart 672 \tl_new:N\l_hyp_current_dest_name_tl
\hyper@anchorend 673 \hook_new:n{hyp/anchor}
hyp/anchor 674 \cs_new_protected:Npn \hyper@anchor #1
\l_hyp_current_dest_name_tl 675 {
676   \exp_args:NnV
677   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
678   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
679   \hook_use:n{hyp/anchor}
680 }
681
682 \cs_new_protected:Npn \hyper@anchorstart #1
683 {
684   \Hy@activeanchortrue
685   \exp_args:NnV
686   \__hyp_destination:nn {#1} \l__hyp_dest_pdfview_tl
687   \tl_set:Nn \l_hyp_current_dest_name_tl {#1}
688   \hook_use:n{hyp/anchor}
689 }
690
691 \cs_new_protected:Npn \hyper@anchorend
692 {
693   \Hy@activeanchorfalse
694 }

```

(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:

```
695 \cs_new_protected:Npn \__hyp_link_goto_begin:nw #1
696   {
697     \mode_leave_vertical:
698     \protected@edef \l__hyp_dest_name_tmpa_tl { #1 }
699     \tl_if_empty:NTF \l__hyp_dest_name_tmpa_tl
700       {
701         \msg_warning:nne
702           { hyp }
703           { empty-destination-name }
704           { \c__hyp_dest_undefined_tl }
705         \tl_set_eq:NN \l__hyp_dest_name_tmpa_tl \c__hyp_dest_undefined_tl
706       }
707       {
708         \__hyp_text_pdfstring:eon
709         { \exp_args:No \HyperDestNameFilter { \l__hyp_dest_name_tmpa_tl } }
710         { \l__hyp_text_enc_dest_tl }
711         \l__hyp_dest_name_tmpa_tl
712       }
713     \exp_args:No
714     \pdfannot_link_goto_begin:nw { \l__hyp_dest_name_tmpa_tl }
715   }
716
717 \cs_new_protected:Npn \__hyp_link_goto_end:
718   {
719     \pdfannot_link_goto_end:
720   }
```

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

```

721
722 \cs_new_protected:Npn \hyper@link #1 #2 #3 %#1 context, #2=destination name, #3 content
723 {
724   \bool_if:NTF \l__hyp_annot_GoTo_bool
725   {
726     \int_gincr:N\g__hyp_linknestlevel_int
727     \__hyp_check_link_nesting:TF
728     {
729       \Hy@VerboseLinkStart{#1}{#2}
730       \group_begin:
731       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}

```

this socket adds something to the /Contents key.

```

732       \socket_use:nmn{hyp/link/GoTo/Contents}{#2}{#3}
733       \hook_use:n {hyp/link/#1}
734       \__hyp_link_goto_begin:nw {#2}#3\Hy@xspace@end
735       \__hyp_link_goto_end:
736       \group_end:
737       \Hy@VerboseLinkStop
738     }
739     {
740       \group_begin: #3\group_end:
741     }
742     \int_gdecr:N\g__hyp_linknestlevel_int
743   }
744   {\let\protect\relax#3}}
745 }
746 \cs_new_protected:Npn \hyper@linkstart #1 #2 %#1 context, #2=destination name
747 {
748   \bool_if:NT \l__hyp_annot_GoTo_bool
749   {
750     \int_gincr:N\g__hyp_linknestlevel_int
751     \__hyp_check_link_nesting:TF
752     {
753       \Hy@VerboseLinkStart{#1}{#2}% only for debug
754       \group_begin:
755       \tl_set:Nn \l_hyp_current_dest_name_tl {#2}
756       \socket_use:nmn{hyp/link/GoTo/Contents}{#2}{#3}
757       \hook_use:n {hyp/link/#1}
758       \__hyp_link_goto_begin:nw {#2}
759     }
760     {
761       \group_begin:
762     }

```

```

763     }
764   }
765
766   \cs_new_protected:Npn \hyper@linkend
767   {
768     \bool_if:NT \l__hyp_annot_GoTo_bool
769     {
770       \__hyp_check_link_nesting:TF
771       {
772         \__hyp_link_goto_end:
773         \group_end:
774         \Hy@VerboseLinkStop
775       }
776       {
777         \group_end:
778       }
779     }
780   }
781 }

```

10.4 URI links

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

```

782 \pdfdict_new:n {l_hyp/annot/A/URI}
783 \pdfdict_put:nnn {l_hyp/annot/A/URI}{Type}{/Action}
784 \pdfdict_put:nnn {l_hyp/annot/A/URI}{S}{/URI}
785
786 \cs_new_protected:Npn \hyper@linkurl #1 #2 %#1:link text #2: URI,
787 {
788   \bool_if:NTF \l__hyp_annot_URI_bool
789   {
790     \int_gincr:N\g__hyp_linknestlevel_int
791     \__hyp_check_link_nesting:TF
792     {
793       \group_begin:
794       \__hyp_text_pdfstring:eoN
795       { #2}
796       { \l__hyp_text_enc_uri_print_tl }
797       \l__hyp_uri_tmpa_tl
798       \pdfdict_put:nno{l_hyp/annot/A/URI}{URI}{\l__hyp_uri_tmpa_tl}
799       \bool_if:NT \l__hyp_href_url_ismap_bool
800       {
801         \pdfdict_put:nnn{l_hyp/annot/A/URI}{IsMap}{true}
802       }
803     }
804   }
805 }

```

This socket adds something to the /Contents key.

```

803     \socket_use:nn{hyp/link/URI/Contents}{#2}
804     \cs_set_eq:NN \# \c_hash_str
805     \cs_set_eq:NN \% \c_percent_str
806     \Hy@safe@activetrue
807     \mode_leave_vertical:
808     \pdfannot_dict_put:nne {link/URI}{A}{<<\pdfdict_use:n {l_hyp/annot/A/URI}>>}
809     \pdfannot_link:nen { URI }
810     {

```

```

811     }
812     {
813     \let\protect\relax
814     #1
815     \Hy@xspace@end
816     \Hy@VerboseLinkStop %where is the start??
817     }
818     \group_end:
819     }
820     {
821     \group_begin: #1 \group_end:
822     }
823     \int_gdecr:N\g__hyp_linknestlevel_int
824     }
825     {\let\protect\relax#1}}
826 }
827

```

10.5 GoToR Links files

```

828 \pdfdict_new:n {l_hyp/annot/A/GoToR}
829 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{Type}{/Action}
830 \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{S}{/GoToR}
831
832 \cs_generate_variant:Nn \pdffile_embed_file:nnn {noe}
833 \cs_new_protected:Npn \hyper@linkfile #1 #2 #3 % link text, filename, destname
834 {
835     \bool_if:NTF \l__hyp_annot_GoToR_bool
836     {
837         \int_gincr:N\g__hyp_linknestlevel_int
838         \__hyp_check_link_nesting:TF
839         {
840             \group_begin:
841             \tl_set:Ne \l__hyp_filename_tmpa_tl { \text_expand:n { #2 } }
842             \exp_args:Ne
843             \pdf_object_if_exist:nF { __hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
844             {
845                 \pdfdict_put:nne { l_pdffile/Filespec}{Subtype}{\pdf_name_from_unicode_e:n
846                 \pdffile_embed_file:noe
847                 {}
848                 {\l__hyp_filename_tmpa_tl }
849                 {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl }
850             }
851             \pdfdict_put:nne
852             {l_hyp/annot/A/GoToR}
853             {F}
854             {\pdf_object_ref:e {__hyp_file_\tl_to_str:N \l__hyp_filename_tmpa_tl}}
855             \__hyp_text_pdfstring:nnN
856             { #3 }
857             { \l__hyp_text_enc_dest_print_tl }
858             \l__hyp_dest_name_tmpa_tl

```

This socket adds something to the /Contents key.

```

859     \socket_use:nn{hyp/link/GoToR/Contents}{#2}

```

```

860     \tl_if_blank:eTF {#3}
861     {
862         \pdfdict_put:nne {l_hyp/annot/A/GoToR}{D}
863         {
864             [
865                 \int_eval:n
866                 { \int_max:nn {0}{ 0\l__hyp_href_pdf_page_tl - 1 }}
867                 /\l__hyp_dest_pdfremotestartview_tl
868             ]
869         }
870     }
871     {
872         \pdfdict_put:nno {l_hyp/annot/A/GoToR}{D}{\l__hyp_dest_name_tmpa_tl}
873     }
874     \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>

```

875     \pdf_object_unnamed_write:ne{dict}{\pdfdict_use:n {l_hyp/annot/A/GoToR}}
876     \pdfannot_dict_put:nne {link/GoToR}{A}{\pdf_object_ref_last:}
877     \pdfannot_link:nnn %expansion??
878     { GoToR }
879     {
880     }
881     {
882         \let\protect\relax
883         #1\Hy@xspace@end
884         \Hy@VerboseLinkStop %where is the start??
885     }
886     \group_end:
887     }
888     {
889         \group_begin: #1 \group_end:
890     }
891     \int_gdecr:N\g__hyp_linknestlevel_int
892     }
893     {{\let\protect\relax#1}}
894     }

```

10.6 Launch links

We define `\hyper@linklaunch` for naming consistency

```

895 \pdfdict_new:n {l_hyp/annot/A/Launch}
896 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{Type}{/Action}
897 \pdfdict_put:nnn {l_hyp/annot/A/Launch}{S}{/Launch}
898
899 \cs_new_protected:Npn \hyper@linklaunch #1 #2 #3 % filename, link text, Parameters
900 {
901     \bool_if:NTF \l__hyp_annot_Launch_bool
902     {
903         \int_gincr:N\g__hyp_linknestlevel_int
904         \__hyp_check_link_nesting:TF
905         {
906             \group_begin:
907             \__hyp_text_pdfstring:nnN

```

```

908         { #1 }
909         { \l__hyp_text_enc_file_print_tl }
910         \l__hyp_filename_tmpa_tl
911         \pdfdict_put:nno {l_hyp/annot/A/Launch}{F}{\l__hyp_filename_tmpa_tl}
912         \__hyp_text_pdfstring:noN
913         { #3 }
914         { \l__hyp_text_enc_para_print_tl }
915         \l__hyp_para_tmpa_tl
916         \bool_if:nTF
917         {
918             \str_if_eq_p:Vn \l__hyp_para_tmpa_tl {}()
919             ||
920             \pdf_version_compare_p:Nn > {1.9}
921         }
922         {
923             \pdfdict_remove:nn {l_hyp/annot/A/Launch}{Win}
924         }
925         {
926             \pdfdict_put:nne
927             {l_hyp/annot/A/Launch}
928             {Win}
929             {<</P \l__hyp_para_tmpa_tl /F \l__hyp_filename_tmpa_tl >>}
930         }
931         \mode_leave_vertical:
932         \pdfannot_dict_put:nne {link/Launch}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Launch}
933         \pdfannot_link:nen
934         { Launch }
935         {
936             % /A
937             % <<
938             % \pdfdict_use:n {l_hyp/annot/A/Launch}
939             % >>
940         }
941         {
942             \let\protect\relax
943             #2\Hy@xspace@end
944             \Hy@VerboseLinkStop %where is the start??
945         }
946         \group_end:
947         }
948         { \group_begin: #2 \group_end: }
949         \int_gdecr:N\g__hyp_linknestlevel_int
950     }
951     {\let\protect\relax#2}}
952 }

```

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

```

953 \use:e
954 { % filename, anchor text, linkname
955     \cs_set_protected:Npn \exp_not:N \@hyper@launch run \c_colon_str #1 \exp_not:N \ \ #2 #3
956 }
957 {
958     \hyper@linklaunch {#1}{#2}{#3}

```



```
959 }
```

10.7 Named links (menu)

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

```
960 \pdfdict_new:n {l_hyp/annot/A/Named}
961 \pdfdict_put:nnn {l_hyp/annot/A/Named}{Type}{/Action}
962 \pdfdict_put:nnn {l_hyp/annot/A/Named}{S}{/Named}
963
964 \cs_new_protected:Npn \hyper@linknamed #1 #2 %#1 action, #2 link text
965 {
966   \bool_if:NTF \l__hyp_annot_Named_bool
967   {
968     \int_gincr:N\g__hyp_linknestlevel_int
969     \__hyp_check_link_nesting:TF
970     {
971       \group_begin:
972       \pdfmeta_standard_verify:nnTF {named_actions}{#1}
973       {
974         \mode_leave_vertical:
975         \pdfdict_put:nne {l_hyp/annot/A/Named}{N}
976           {\pdf_name_from_unicode_e:n{#1}}
977         \pdfannot_dict_put:nne {link/Named}{A}{<<\pdfdict_use:n {l_hyp/annot/A/Named}
978         \pdfannot_link:nnn { Named }
979         {
980           % /A
981           % <<
982           % \pdfdict_use:n { l_hyp/annot/A/Named }
983           % >>
984         }
985         {
986           #2
987           \Hy@xspace@end
988           \Hy@VerboseLinkStop
989         }
990       }
991       {
992         \msg_warning:nnn { hyp } { pdfa-no-named-action }{#1}
993         #2
994       }
995       \group_end:
996     }
997     { \group_begin: #2 \group_end: }
998     \int_gdecr:N\g__hyp_linknestlevel_int
999   }
1000   {{\let\protect\relax#2}}
1001 }
1002
```

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.

```
1003 \cs_generate_variant:Nn \color_export:nnN {nVN}
```

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

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

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

```
1004 \cs_new_protected:Npn \__hyp_color_select:n #1
1005 {
1006   \tl_if_head_eq_charcode:nNTF {#1}[ %]
1007   {
1008     \__hyp_color_select_aux:wn #1
1009   }
1010   {
1011     \color_select:n {#1}
1012   }
1013 }
1014
1015 \cs_new_protected:Npn \__hyp_color_select_aux:wn [#1] #2
1016 {
1017   \color_select:nn {#1}{#2}
1018 }
1019
1020 \cs_generate_variant:Nn \__hyp_color_select:n {e}
```

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

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

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

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

```
1021 \cs_new_protected:Npn \__hyp_color_set:nn #1 #2
1022 {
1023   \tl_if_head_eq_charcode:nNTF {#2}[ %]
1024   {
1025     \__hyp_color_set_aux:nwn { #1 } #2
1026   }
1027   {
1028     \color_set:nn {#1} {#2}
1029   }
}
```

```

1030 }
1031
1032 \cs_new_protected:Npn \__hyp_color_set_aux:nwn #1 [#2] #3
1033 {
1034   \color_set:nnn {#1}{#2}{#3}
1035 }
1036
1037 \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.

```

1038 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1039 {
1040   \hook_gput_code:nnn
1041     {pdfannot/link/#2/begin}
1042     {hyp/color}
1043     {
1044       \bool_if:cT { l_hyp_annot_color#1_bool }
1045       {
1046         \group_begin:
1047         \color_select:n { hyp/color/#1}
1048       }
1049     }
1050   \hook_gput_code:nnn
1051     {pdfannot/link/#2/end}
1052     {hyp/color}
1053     {
1054       \bool_if:cT { l_hyp_annot_color#1_bool }
1055       {
1056         \group_end:
1057       }
1058     }
1059 }

```

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

```

1060 \keys_define:nn { hyp }
1061 {
1062   ,colorlinks .choice:
1063   ,colorlinks / true .meta:n =
1064   {
1065     ,pdfborder={0~0~0}
1066     ,pdfborderstyle=
1067     ,colorurl =#1
1068     ,colorlink =#1
1069     ,colorryn =#1
1070     ,colormenu =#1
1071     ,colorfile =#1
1072   }
1073   ,colorlinks / false .meta:n =

```

```

1074     {
1075         ,colorurl =#1
1076         ,colorlink =#1
1077         ,colorrund =#1
1078         ,colormenu =#1
1079         ,colorfile =#1
1080     }
1081     ,colorlinks .default:n = {true}
1082 }

colorurl (setup key)
colorlink (setup key) 1083 \seq_map_inline:Nn \c__hyp_annot_types_seq
colorrund (setup key) 1084 {
colormenu (setup key) 1085     \keys_define:nn { hyp }
colorfile (setup key) 1086     {
urlcolor (setup key) 1087         ,color#1 .bool_set:c = { l_hyp_annot_color#1_bool }
linkcolor (setup key) 1088         ,#1color .code:n = { \__hyp_color_set:ne {hyp/color/#1}{##1} }
runcolor (setup key) 1089     }
menucolor (setup key) 1090 }
filecolor (setup key) 1091 \keys_define:nn { hyp }
allcolors (setup key) 1092 {
1093     ,allcolors .meta:n =
1094     {
1095         ,urlcolor=#1
1096         ,linkcolor=#1
1097         ,runcolor=#1
1098         ,filecolor=#1
1099         ,menucolor=#1
1100     }
1101     ,allcolors .value_required:n = true
1102 }
1103 }

```

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)

```

1104 \keys_define:nn { hyp }
1105 {
1106     ,bordercolormodel .choices:nn =
1107     {rgb,cmyk}
1108     { \str_gset:Nn \g__hyp_bordercolormodel_str {space-sep-#1}}
1109     ,bordercolormodel .initial:n ={rgb}
1110 }

1111 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1112 {
1113     \keys_define:nn { hyp }
1114     {
1115         #1bordercolor .code:n =

```

```

1116     {
1117         \tl_if_empty:nTF { ##1 }
1118         {
1119             \pdfannot_dict_remove:nn
1120             {link/#2}
1121             { C }
1122         }
1123         {
1124             \__hyp_color_set:ne {hyp/color/#1border}{##1}
1125             \color_export:nVN
1126             {hyp/color/#1border}
1127             \g__hyp_bordercolormodel_str
1128             \l__hyp_tmpa_tl
1129             \pdfannot_dict_put:nne
1130             {link/#2}
1131             { C }
1132             { [\l__hyp_tmpa_tl] }
1133         }
1134     }
1135 }
1136 }
1137
1138 \keys_define:nn { hyp }
1139 {
1140     ,allbordercolors .meta:n =
1141     {
1142         ,linkbordercolor=#1
1143         ,urlbordercolor =#1
1144         ,filebordercolor=#1
1145         ,menubordercolor=#1
1146         ,runbordercolor =#1
1147     }
1148     ,allbordercolors .value_required:n = true
1149 }
1150

```

11.3.2 Borderwidth and -arc

```

1151 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1152 {
1153     \keys_define:nn { hyp }
1154     {
1155         #1border .code:n =
1156         {
1157             \tl_if_empty:nTF { ##1 }
1158             {
1159                 \pdfannot_dict_remove:nn
1160                 {link/#2}
1161                 { Border }
1162             }
1163             {
1164                 \pdfannot_dict_put:nnn
1165                 {link/#2}
1166                 { Border }

```

```

1167         { [#1] }
1168     }
1169 }
1170 }
1171 }
1172 \keys_define:nn { hyp }
1173 {
1174   ,pdfborder .code:n =
1175   {
1176     \tl_if_empty:nTF { #1 }
1177     {
1178       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1179       {
1180         \pdfannot_dict_remove:nn
1181         {link/##2}
1182         { Border }
1183       }
1184     }
1185     {
1186       \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1187       {
1188         \pdfannot_dict_put:nnn
1189         {link/##2}
1190         { Border }
1191         { [#1] }
1192       }
1193     }
1194   }
1195   ,pdfborder .initial:n = {0~0~1},
1196 }

```

11.3.3 Borderstyle

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

```

pdfborderstyle (setup key)
urlborderstyle (setup key) 1197 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
linkborderstyle (setup key) 1198 {
runborderstyle (setup key) 1199   \keys_define:nn { hyp }
fileborderstyle (setup key) 1200   {
menuborderstyle (setup key) 1201     #1borderstyle .code:n =
1202     {
1203       \tl_if_empty:nTF { ##1 }
1204       {
1205         \pdfannot_dict_remove:nn
1206         {link/#2}
1207         { BS }
1208       }
1209       {
1210         \pdfannot_dict_put:nnn
1211         {link/#2}
1212         { BS }
1213         { <<##1>> }
1214       }
1215     }

```

```

1216     }
1217   }
1218   \keys_define:nn { hyp }
1219   {
1220     ,pdfborderstyle .code:n =
1221     {
1222       \tl_if_empty:nTF { #1 }
1223       {
1224         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1225         {
1226           \pdfannot_dict_remove:nn
1227             {link/##2}
1228             { BS }
1229         }
1230       }
1231       {
1232         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1233         {
1234           \pdfannot_dict_put:nnn
1235             {link/##2}
1236             { BS }
1237             { <<#1>> }
1238         }
1239       }
1240     }
1241     ,pdfborderstyle .initial:n = {},
1242   }

```

11.4 ocolorlinks

OCG colorlinks need objects and an entry in the catalog. Perhaps the objects need public names to avoid that ocolor2 has to create duplicates? TODO

`__hyp_ocg_init:` This commands write the objects as needed if ocolor links are used. The initialization should happens only once.

```

1243 \cs_new_protected:Npn \__hyp_ocg_init:
1244 {
1245   \pdf_object_new:n { __hyp/OCG/View }
1246   \pdf_object_new:n { __hyp/OCG/Print }
1247   \pdf_object_new:n { __hyp/OCG/config }
1248   \pdf_object_new:n { __hyp/OCG/refarray }
1249   \pdf_object_write:nne { __hyp/OCG/refarray } { array }
1250   {
1251     \pdf_object_ref:n { __hyp/OCG/View }
1252     \c_space_tl
1253     \pdf_object_ref:n { __hyp/OCG/Print }
1254   }
1255   \pdf_object_write:nnn { __hyp/OCG/View } { dict }
1256   {
1257     /Type/OCG
1258     /Name(View)
1259     /Usage
1260     <<
1261     /Print <</PrintState/OFF>>~

```

```

1262         /View <</ViewState/ON >>~
1263     >>
1264 }
1265 \pdf_object_write:nnn { __hyp/OCG/Print } { dict }
1266 {
1267     /Type/OCG
1268     /Name(Print)
1269     /Usage
1270     <<
1271         /Print <</PrintState/ON>>~
1272         /View <</ViewState/OFF>>~
1273     >>
1274 }
1275 \pdfmanagement_add:nne { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC
1276 \pdfmanagement_add:nne { Catalog / OCProperties }{OCGs }{ \pdf_object_ref:n {__hyp/OC
1277 \pdf_object_write:nne { __hyp/OCG/config } { dict }
1278 {
1279     /OFF[\pdf_object_ref:n { __hyp/OCG/Print } ]
1280     /AS[
1281         <<
1282             /Event/View
1283             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1284             /Category[/View]
1285         >>
1286         <<
1287             /Event/Print
1288             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1289             /Category[/Print]
1290         >>
1291         <<
1292             /Event/Export
1293             /OCGs\c_space_tl \pdf_object_ref:n { __hyp/OCG/refarray }
1294             /Category[/Print]
1295         >>
1296     ]
1297 }
1298 \pdfmanagement_add:nne { Catalog / OCProperties }{ D }{ \pdf_object_ref:n { __hyp/OCG
1299 \cs_gset:Npn \__hyp_ocg_init: {}
1300 }

```

(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.

```

1301 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1302 {
1303     \hook_gput_code:nnn
1304     {pdfannot/link/#2/begin}
1305     {hyp/ocg}
1306     {
1307         \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1308         {
1309             \__hyp_ocg_init:
1310             \group_begin:
1311             \hbox_set:Nw \l__hyp_tmpa_box

```



```

1312     }
1313   }
1314   \hook_gput_code:nnn
1315     {pdfannot/link/#2/end}
1316     {hyp/ocg}
1317     {
1318       \bool_if:cT { l_hyp_annot_ocgcolor#1_bool }
1319       {
1320         \hbox_set_end:
1321         \mbox
1322         {
1323           \pdf_bdcobject:nn {OC}{__hyp/OCG/Print}
1324           \hbox_overlap_right:n { \box_use:N \l__hyp_tmpa_box }
1325           \pdf_emc:
1326           \pdf_bdcobject:nn {OC}{__hyp/OCG/View}
1327           \group_begin:
1328           \color_select:n { hyp/color/#1 }
1329           \box_use_drop:N \l__hyp_tmpa_box
1330           \group_end:
1331           \pdf_emc:
1332         }
1333         \group_end:
1334       }
1335     }
1336   }

```

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

```

ocgcolorlink (setup key) 1337 \bool_lazy_or:nnTF
ocgcolorurl (setup key) 1338 { \pdf_version_compare_p:Nn > {1.4} }
ocgcolorfile (setup key) 1339 { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
ocgcolormenu (setup key) 1340 {
ocgcolorrun (setup key) 1341   \keys_define:nn { hyp }
1342   {
1343     ,_ocgcolorlinks .meta:n =
1344     {
1345       ocgcolorlink=#1,
1346       ocgcolorurl=#1,
1347       ocgcolorfile=#1,
1348       ocgcolorrun=#1,
1349       ocgcolormenu=#1
1350     }
1351     ,_ocgcolorlinks .default:n = true
1352   }
1353 }
1354 {
1355   \keys_define:nn { hyp }
1356   {
1357     ,_ocgcolorlinks .code:n =
1358     {
1359       \msg_warning:nnee
1360       { hyp }
1361       { ignore-deprecated-or-unknown-option-in-pdf-version }
1362       { ocgcolorlinks } { \pdf_version_major:.\pdf_version_minor: }
1363     }

```

```

1364     }
1365   }
1366
1367   \keys_define:nn { hyp }
1368   {
1369     ,ocgcolorlinks .choice:
1370     ,ocgcolorlinks / true .meta:n =
1371     {
1372       pdfborder      = {0~0~0},
1373       pdfborderstyle = {},
1374       colorlinks     = false,
1375       _ocgcolorlinks = true
1376     }
1377     ,ocgcolorlinks / false .meta:n =
1378     {
1379       _ocgcolorlinks = false
1380     }
1381     ,ocgcolorlinks .default:n = {true}
1382   }
1383
1384   \seq_map_inline:Nn \c__hyp_annot_types_seq
1385   {
1386     \bool_lazy_or:nnTF
1387     { \pdf_version_compare_p:Nn > {1.4} }
1388     { \str_if_eq_p:ee{\pdf_version_major:}{-1} }
1389     {
1390       \keys_define:nn { hyp }
1391       {
1392         ,ocgcolor#1 .bool_set:c = { l_hyp_annot_ocgcolor#1_bool }
1393       }
1394     }
1395     {
1396       \keys_define:nn { hyp }
1397       {
1398         ,ocgcolor#1 .code:n=
1399         {
1400           \msg_warning:nnee
1401           { hyp }
1402           { ignore-deprecated-or-unknown-option-in-pdf-version }
1403           { ocgcolor#1 }
1404           { \pdf_version_major:.\pdf_version_minor: }
1405         }
1406       }
1407     }
1408   }

```

11.5 Highlighting

This keys set what happens if you click on a link

```

1409   \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1410   {
1411     \keys_define:nn { hyp }
1412     {
1413       ,#1highlight .choices:nn =

```

```

1414     { /I, /N, /O, /P}
1415     {
1416         \pdfannot_dict_put:nnn
1417         {link/#2}
1418         { H }
1419         { ##1 }
1420
1421     }
1422     ,#1highlight / .code:n =
1423     {
1424         \pdfannot_dict_remove:nn
1425         {link/#2}
1426         { H }
1427
1428     }
1429     ,#1highlight / unknown .code:n =
1430     {
1431         \msg_warning:nneee { hyp } { unknown-choice+empty }
1432         { #1highlight }
1433         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }
1434         { \exp_not:n {##1} }
1435     }
1436 }
1437 }
1438
1439
1440 \keys_define:nn { hyp }
1441 {
1442     ,pdfhighlight .choices:nn =
1443     { /I, /N, /O, /P}
1444     {
1445         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1446         {
1447             \pdfannot_dict_put:nnn
1448             {link/###2}
1449             { H }
1450             { #1 }
1451         }
1452     }
1453     ,pdfhighlight / .code:n =
1454     {
1455         \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
1456         {
1457             \pdfannot_dict_remove:nn
1458             {link/##2}
1459             { H }
1460         }
1461     }
1462     ,pdfhighlight .initial:n = {/I},
1463     ,pdfhighlight / unknown .code:n =
1464     {
1465         \msg_warning:nneee { hyp } { unknown-choice+empty }
1466         { pdfhighlight }
1467         { /I~(inverse), /N~(no effect), /O~(outline), /P~(inset) }

```

```

1468         { \exp_not:n {#1} }
1469     }
1470 }

```

11.6 Hiding links

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

```

hidelinks (setup key)
hidelink (setup key) 1471 \keys_define:nn { hyp }
hideurl (setup key) 1472 {
hidefile (setup key) 1473     hidelinks .meta:n =
hiderun (setup key) 1474     {
hidemenu (setup key) 1475         ,colorlinks      = false
1476         ,ocgcolorlinks = false
1477         ,pdfborder      = { 0~0~0 }
1478         ,pdfborderstyle=
1479     }
1480 }
1481
1482 \seq_map_inline:Nn \c__hyp_annot_types_seq
1483 {
1484     \keys_define:nn { hyp }
1485     {
1486         hide#1 .meta:n =
1487         {
1488             ,color#1      = false
1489             ,ocgcolor#1  = false
1490             ,#1border     = { 0~0~0 }
1491             ,#1borderstyle =
1492         }
1493     }
1494 }

```

11.7 color schemes and settings

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

```

colorscheme (setup key)
1495 \keys_define:nn { hyp }
1496 {
1497     colorscheme .code:n =
1498     {
1499         \prop_map_inline:cn { c__hyp_colorscheme_#1_prop }
1500         {
1501             \keys_set:nn { hyp }
1502             {
1503                 ##1 = ##2
1504             }
1505         }
1506     }
1507 }
1508 \keys_set:nn { hyp } {colorscheme=phelype}

```

12 Keys

12.1 Ignored keys

The following are ignored (with or without warnings)

```
unicode (setup key)
pdfencoding (setup key) 1509 \keys_define:nn { hyp }
pdfversion (setup key) 1510 {
1511     ,unicode .code:n = {}
1512     ,pdfencoding .code:n = {}
1513     ,pdfversion .code:n =
1514     {
1515         \msg_warning:nn { hyp }{ pdfversion-disabled }
1516     }
1517 }
1518 %
```

12.2 Various keys for the pdf and linking behaviour

This keys are typically set only once.

```
verbose (setup key)
debug (setup key) 1519 \keys_define:nn { hyp }
draft (setup key) 1520 {
final (setup key) 1521     ,verbose .legacy_if_set:n = {Hy@verbose}
1522     ,debug .legacy_if_set:n = {Hy@verbose}
1523 }
1524 \keys_define:nn { hyp }
1525 {
1526     ,draft .code:n =
1527     {
1528         \Hy@drafttrue
1529         \PassOptionsToPackage{draft}{bookmark}
1530     }
1531     ,final .code:n =
1532     {
1533         \Hy@finaltrue
1534         \PassOptionsToPackage{final}{bookmark}
1535     }
1536 }

extension (setup key)
hypertextnames (setup key) 1537 \keys_define:nn { hyp }
naturalnames (setup key) 1538 {
pageanchor (setup key) 1539     ,extension .tl_set:N = \XR@ext
linktoc (setup key) 1540     ,extension .initial:n= pdf
linktocpage (setup key) 1541     ,hypertextnames .legacy_if_set:n = {Hy@hypertextnames}
plainpages (setup key) 1542     ,linkfileprefix .tl_set:N = \Hy@linkfileprefix
localanchorname (setup key) 1543     ,localanchorname .legacy_if_set:n = {Hy@localanchorname}
linkfileprefix (setup key) 1544     ,naturalnames .legacy_if_set:n = {Hy@naturalnames}
1545     ,pageanchor .legacy_if_set:n = {Hy@pageanchor}
1546     ,plainpages .legacy_if_set:n = {Hy@plainpages}
1547 }
```

```

1548
1549 \keys_define:nn { hyp }
1550 {
1551   ,linktoc .choices:nn = { none, section, all, page }
1552   {
1553     \cs_set_eq:Nc \Hy@linktoc { Hy@linktoc@#1 }
1554   }
1555   ,linktoc / unknown .code:n =
1556   {
1557     \msg_warning:nree { hyp } { unknown-choice }
1558     { linktoc }
1559     { none, section, all, page }
1560     { \exp_not:n {#1} }
1561   }
1562   ,linktocpage .choice:
1563   ,linktocpage / true .meta:n = {linktoc=page}
1564   ,linktocpage / false .meta:n = {linktoc=section}
1565   ,linktocpage .default:n = true
1566 }
1567

```

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

```

url (setup key) 1568 \prop_map_inline:Nn \c__hyp_map_hyp_annot_prop
file (setup key) 1569 {
menu (setup key) 1570   \keys_define:nn { hyp }
run (setup key) 1571   {
1572     ,#1 .bool_set:c = {l__hyp_annot_#2_bool}
1573   }
1574 }

```

```

1575 \keys_define:nn { hyp }
1576 {
1577   ,baseurl .code:n =
1578   {
1579     \__hyp_text_pdfstring:ooN { #1 } {\l__hyp_text_enc_uri_print_tl} \l__hyp_tmpa_tl
1580     \tl_if_empty:NTF \l__hyp_tmpa_tl
1581     {
1582       \pdfmanagement_remove:nn {Catalog} { URI }
1583     }
1584     {
1585       \pdfmanagement_add:nne {Catalog} { URI }{ <</Base \l__hyp_tmpa_tl>> }
1586     }
1587     \__hyp_store_metadata:nn {baseurl}{#1}
1588   }
1589   %only false does something ...
1590   ,bookmarks .choice:
1591   ,bookmarks / false .code:n = {\RemoveFromHook {begindocument/before}[hyperref/bookmark]}
1592   ,bookmarks / true .code:n = {}
1593   ,bookmarks .default:n = {true}
1594   ,bookmarksnumbered .legacy_if_set:n = {Hy@bookmarksnumbered}
1595   ,bookmarksopen .legacy_if_set:n = {Hy@bookmarksopen}
1596   ,bookmarksopenlevel .tl_set:N = \@bookmarksopenlevel
1597   ,bookmarkstype .tl_set:N = \Hy@bookmarkstype
1598   ,pdfcenterwindow .choice:

```

```

1599 ,pdfcenterwindow / false .code:n =
1600 {
1601   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1602 }
1603 ,pdfcenterwindow / true .code:n =
1604 {
1605   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { CenterWindow }{ true }
1606 }
1607 ,pdfcenterwindow / .code:n =
1608 {
1609   \pdfmanagement_remove:nn {Catalog / ViewerPreferences }{ CenterWindow }
1610 }
1611 ,pdfcenterwindow / unknown .code:n =
1612 {
1613   \msg_warning:nnee { hyp } { no-bool }
1614   { pdfcenterwindow }
1615   { \exp_not:n {#1} }
1616 }
1617 ,pdfcenterwindow .default:n = true
1618 ,pdfdirection .choice:
1619 ,pdfdirection / L2R .code:n =
1620 {
1621   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /L2R }
1622 }
1623 ,pdfdirection / R2L .code:n =
1624 {
1625   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { Direction }{ /R2L }
1626 }
1627 ,pdfdirection / .code:n =
1628 {
1629   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Direction }
1630 }
1631 ,pdfdirection / unknown .code:n =
1632 {
1633   \msg_warning:nneee { hyp } { unknown-choice+empty }
1634   { pdfdirection }
1635   { L2R , R2L }
1636   { \exp_not:n {#1} }
1637 }
1638 ,pdfdisplaydoctitle .choice:
1639 ,pdfdisplaydoctitle / false .code:n =
1640 {
1641   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { DisplayDocTitle }
1642 }
1643 ,pdfdisplaydoctitle / true .code:n =
1644 {
1645   \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
1646 }
1647 ,pdfdisplaydoctitle .default:n = true
1648 ,pdfduplex .choices:nn =
1649 {Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge}
1650 {
1651   \pdf_version_compare:NnTF > {1.6}
1652   {

```

```

1653         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1654         { Duplex } { /#1 }
1655     }
1656     {
1657         \msg_warning:nnee
1658         {hyp}
1659         {ignore-deprecated-or-unknown-option-in-pdf-version}
1660         {pdfduplex}
1661         {\pdf_version:}
1662     }
1663 }%
1664 ,pdfduplex / .code:n =
1665 {
1666     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { Duplex }
1667 }
1668 ,pdfduplex / unknown .code:n =
1669 {
1670     \msg_warning:nneee { hyp } { unknown-choice+empty }
1671     { pdfduplex }
1672     { Simplex, DuplexFlipShortEdge, DuplexFlipLongEdge }
1673     { \exp_not:n {#1} }
1674 }
1675 ,pdffitwindow .choice:
1676 ,pdffitwindow / false .code:n =
1677 {
1678     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1679 }
1680 ,pdffitwindow / true .code:n =
1681 {
1682     \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { FitWindow } { true }
1683 }
1684 ,pdffitwindow / .code:n =
1685 {
1686     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { FitWindow }
1687 }
1688 ,pdffitwindow .default:n = true
1689 ,pdffitwindow / unknown .code:n =
1690 {
1691     \msg_warning:nnee { hyp } { no-bool }
1692     { pdffitwindow }
1693     { \exp_not:n {#1} }
1694 }
1695 ,pdflinkmargin .code:n = { \pdfannot_link_margin:n { #1 } }
1696 ,pdflinkmargin .initial:n = {1pt}
1697 ,pdfmenubar .choice:
1698 ,pdfmenubar / true .code:n =
1699 {
1700     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1701 }
1702 ,pdfmenubar / false .code:n =
1703 {
1704     \pdfmanagement_add:nn {Catalog / ViewerPreferences }
1705     { HideMenubar } { true }
1706 }

```



```

1707 ,pdfmenubar / .code:n =
1708 {
1709     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideMenubar }
1710 }
1711 ,pdfmenubar .default:n = true
1712 ,pdfmenubar / unknown .code:n =
1713 {
1714     \msg_warning:nnee { hyp } { no-bool }
1715     { pdfmenubar }
1716     { \exp_not:n {#1} }
1717 }
1718 ,pdfnewwindow .choice:
1719 ,pdfnewwindow / true .code:n =
1720 {
1721     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{true}
1722     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{true}
1723 }
1724 ,pdfnewwindow / false .code:n =
1725 {
1726     \pdfdict_put:nnn {l_hyp/annot/A/GoToR}{/NewWindow}{false}
1727     \pdfdict_put:nnn {l_hyp/annot/A/Launch}{/NewWindow}{false}
1728 }
1729 ,pdfnewwindow / .code:n =
1730 {
1731     \pdfdict_remove:nn {l_hyp/annot/A/GoToR}{/NewWindow}
1732     \pdfdict_remove:nn {l_hyp/annot/A/Launch}{/NewWindow}
1733 }
1734 ,pdfnonfullscreenpagemode .choices:nn =
1735 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1736 {
1737     \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1738     { NonFullScreenPageMode } {/#1}
1739 }
1740 ,pdfnonfullscreenpagemode / UseAttachments .code:n =
1741 {
1742     \pdf_version_compare:NnTF < {1.6}
1743     {
1744         %message
1745     }
1746     {
1747         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1748         {NonFullScreenPageMode}{/UseAttachments}
1749     }
1750 }
1751 ,pdfnonfullscreenpagemode / .code:n =
1752 {
1753     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NonFullScreenPageMode }
1754 }
1755 ,pdfnonfullscreenpagemode / unknown .code:n =
1756 {
1757     \msg_warning:nneee { hyp } { unknown-choice+empty }
1758     { pdfnonfullscreenpagemode }
1759     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1760     { \exp_not:n {#1} }

```

```

1761     }
1762 ,pdfnumcopies .code:n =
1763     {
1764         \pdf_version_compare:NnTF > {1.6}
1765         {
1766             \tl_if_empty:nTF {#1}
1767             {
1768                 \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { NumCopies }
1769             }
1770             {
1771                 \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1772                 {NumCopies}{#1}
1773             }
1774         }
1775     {
1776         \msg_warning:nnee
1777         {hyp}
1778         {ignore-deprecated-or-unknown-option-in-pdf-version}
1779         {pdfnumcopies}
1780         {\pdf_version:}
1781     }
1782 }
1783 ,pdfpagelayout .choices:nn =
1784 { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight}
1785 { \pdfmanagement_add:nne {Catalog} { PageLayout }{ /#1 } }
1786 ,pdfpagelayout / .code:n =
1787 { \pdfmanagement_remove:nn {Catalog} { PageLayout } }
1788 ,pdfpagelayout / unknown .code:n =
1789 {
1790     \msg_warning:nneee { hyp } { unknown-choice+empty }
1791     { pdfpagelayout }
1792     { SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight }
1793     { \exp_not:n {#1} }
1794 }
1795 ,pdfpagemode .choices:nn =
1796 { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC } %pdf 1.5
1797 { \pdfmanagement_add:nne {Catalog} { PageMode }{ /#1 } }
1798 ,pdfpagemode / UseAttachments .code:n =
1799 {
1800     \pdf_version_compare:NnTF > {1.5}
1801     {
1802         \pdfmanagement_add:nne {Catalog} { PageMode }{ /UseAttachments }
1803     }
1804     {
1805         \msg_warning:nnee
1806         {hyp}
1807         {ignore-deprecated-or-unknown-value-in-pdf-version}
1808         {UseAttachments}
1809         {\pdf_version:}
1810     }
1811 }
1812 ,pdfpagemode .initial:n = { UseOutlines } %for now ...
1813 ,pdfpagemode / unknown .code:n =
1814 {

```

```

1815     \msg_warning:nneee { hyp } { unknown-choice+empty }
1816     { pdfpagemode }
1817     { UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments (PDF 1.6) }
1818     { \exp_not:n {#1} }
1819 }
1820 ,pdfpagescrop .code:n =
1821 {
1822     \tl_if_empty:nTF {#1} %or blank?
1823     {
1824         \pdfmanagement_remove:nn {Pages} { CropBox }
1825     }
1826     {
1827         \pdfmanagement_add:nne {Pages} { CropBox } { [#1] }
1828     }
1829 }
1830 ,pdfpicktraybypdfsize .choice:
1831 ,pdfpicktraybypdfsize / true .code:n =
1832 {
1833     \pdf_version_compare:NnTF > {1.6}
1834     {
1835         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1836         { PickTrayByPDFSize } { true }
1837     }
1838     {
1839         \msg_warning:nnee
1840         {hyp}
1841         {ignore-deprecated-or-unknown-option-in-pdf-version}
1842         {pdfpicktraybypdfsize}
1843         {\pdf_version:}
1844     }
1845 }
1846 ,pdfpicktraybypdfsize / false .code:n =
1847 {
1848     \pdf_version_compare:NnTF > {1.6}
1849     {
1850         \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
1851         { PickTrayByPDFSize } { false }
1852     }
1853     {
1854         \msg_warning:nnee
1855         {hyp}
1856         {ignore-deprecated-or-unknown-option-in-pdf-version}
1857         {pdfpicktraybypdfsize}
1858         {\pdf_version:}
1859     }
1860 }
1861 ,pdfpicktraybypdfsize / .code:n =
1862 {
1863     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PickTrayByPDFSize }
1864 }
1865 ,pdfpicktraybypdfsize / unknown .code:n =
1866 {
1867     \msg_warning:nnee { hyp } { no-bool }
1868     { picktraybypdfsize }

```

```

1869         { \exp_not:n {#1} }
1870     }
1871 ,pdfprintarea .choices:nn =
1872 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1873 {
1874     \pdf_version_compare:NnTF < {2.0}
1875     {
1876         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1877         { PrintArea } { /#1 }
1878     }
1879     {
1880         \msg_warning:nnee
1881         {hyp}
1882         {ignore-deprecated-or-unknown-option-in-pdf-version}
1883         {pdfprintarea}
1884         {\pdf_version:}
1885     }
1886 }%
1887 ,pdfprintarea / .code:n =
1888 { \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintArea } }
1889 ,pdfprintarea / unknown .code:n =
1890 {
1891     \msg_warning:nnee { hyp } { unknown-choice+empty }
1892     { pdfprintarea }
1893     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1894     { \exp_not:n {#1} }
1895 }
1896 ,pdfprintclip .choices:nn =
1897 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
1898 {
1899     \pdf_version_compare:NnTF < {2.0}
1900     {
1901         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1902         { PrintClip } { /#1 }
1903     }
1904     {
1905         \msg_warning:nnee
1906         {hyp}
1907         {ignore-deprecated-or-unknown-option-in-pdf-version}
1908         {pdfprintclip}
1909         {\pdf_version:}
1910     }
1911 }%
1912 ,pdfprintclip / .code:n =
1913 {
1914     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { PrintClip }
1915 }
1916 ,pdfprintclip / unknown .code:n =
1917 {
1918     \msg_warning:nnee
1919     { hyp }
1920     { unknown-choice+empty }
1921     { pdfprintclip }
1922     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }

```

```

1923     { \exp_not:n {#1} }
1924   }
1925 ,pdfprintpagerange .code:n =
1926   {
1927     \pdf_version_compare:NnTF > {1.6}
1928     {
1929       \tl_if_empty:nTF { #1}
1930       {
1931         \pdfmanagement_remove:nn {Catalog / ViewerPreferences }
1932         { PrintPageRange }
1933       }
1934       {
1935         \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1936         {PrintPageRange}{[#1]}
1937       }
1938     }
1939     {
1940       \msg_warning:nnee
1941       {hyp}
1942       {ignore-deprecated-or-unknown-option-in-pdf-version}
1943       {pdfprintpagerange}
1944       {\pdf_version:}
1945     }
1946   }
1947 ,pdfprintscaling .choices:nn =
1948   { None, AppDefault }
1949   {
1950     \pdf_version_compare:NnTF > {1.5}
1951     {
1952       \pdfmanagement_add:nne {Catalog / ViewerPreferences }
1953       { PrintScaling } { /#1 }
1954     }
1955     {
1956       \msg_warning:nnee
1957       {hyp}
1958       {ignore-deprecated-or-unknown-option-in-pdf-version}
1959       {pdfprintscaling}
1960       {\pdf_version:}
1961     }
1962   }%
1963 ,pdfprintscaling / .code:n =
1964   {
1965     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {PrintScaling }
1966   }
1967 ,pdfprintscaling / unknown .code:n =
1968   {
1969     \msg_warning:nnee { hyp } { unknown-choice+empty }
1970     { pdfprintarea }
1971     { None, AppDefault }
1972     { \exp_not:n {#1} }
1973   }
1974 ,pdfremotestartview .code:n =
1975   {
1976     \tl_set:Ne \l__hyp_tmpa_tl {#1-null-null-null-}

```

```

1977     \exp_args:NNV
1978     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
1979     {
1980         \tl_set:Ne \l__hyp_dest_pdfremotestartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
1981     }
1982     {
1983         \msg_warning:nmmm {hyp}{invalid-destination-value}{#1}{pdfremotestartview}
1984         \tl_set:Nn \l__hyp_dest_pdfremotestartview_tl {Fit}
1985     }
1986 }
1987 ,pdfremotestartview .initial:n = {Fit}
1988 % pdfstartpage is special as it shares code with pdfstartview
1989 ,pdfstartpage .code:n =
1990 {
1991     \tl_gset:Ne \g__hyp_dest_pdfstartpage_tl { #1 }
1992     \bool_if:nTF
1993     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
1994     {
1995         \pdfmanagement_remove:nn {Catalog} { OpenAction }
1996     }
1997     {
1998         \pdfmanagement_add:nne {Catalog} { OpenAction }
1999         {
2000             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2001         }
2002     }
2003 }
2004 ,pdfstartpage .initial:n =1
2005 ,pdfstartview .code:n =
2006 {
2007     \tl_set:Ne \l__hyp_tmpa_tl {#1~null~null~null~}
2008     \exp_args:NNV
2009     \regex_extract_once:NnNTF \c__hyp_dest_startview_regex \l__hyp_tmpa_tl \l__hyp_tmpa_
2010     {
2011         \tl_gset:Ne \g__hyp_dest_pdfstartview_tl {\seq_item:Nn \l__hyp_tmpa_seq {1}}
2012     }
2013     {
2014         \msg_warning:nmmm {hyp}{invalid-destination-value}{#1}{pdfstartview}
2015         \tl_gset:Nn \g__hyp_dest_pdfstartview_tl {Fit}
2016     }
2017     \bool_if:nTF
2018     { \tl_if_empty_p:N \g__hyp_dest_pdfstartpage_tl || \tl_if_empty_p:N \g__hyp_dest_pd
2019     {
2020         \pdfmanagement_remove:nn {Catalog} { OpenAction }
2021     }
2022     {
2023         \pdfmanagement_add:nne {Catalog} { OpenAction }
2024         {
2025             [\pdf_pageobject_ref:n {\g__hyp_dest_pdfstartpage_tl}~/\g__hyp_dest_pdfstartv
2026         }
2027     }
2028 }
2029 ,pdfstartview .initial:n = Fit
2030 ,pdftoolbar .choice:

```

```

2031 ,pdftoolbar / true .code:n =
2032 {
2033   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2034 }
2035 ,pdftoolbar / false .code:n =
2036 {
2037   \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2038   { HideToolbar } { true }
2039 }
2040 ,pdftoolbar / true .code:n =
2041 {
2042   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideToolbar }
2043 }
2044 ,pdftoolbar .default:n = true
2045 ,pdftoolbar / unknown .code:n =
2046 {
2047   \msg_warning:nnee { hyp } { no-bool }
2048   { pdftoolbar }
2049   { \exp_not:n {#1} }
2050 }
2051 % pdfview see below.
2052 ,pdfviewarea .choices:nn =
2053 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2054 {
2055   \pdf_version_compare:NnTF < {2.0}
2056   {
2057     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2058     { ViewArea } { /#1 }
2059   }
2060   {
2061     \msg_warning:nnee
2062     {hyp}
2063     {ignore-deprecated-or-unknown-option-in-pdf-version}
2064     {pdfviewarea}
2065     {\pdf_version:}
2066   }
2067 }%
2068 ,pdfviewarea / .code:n =
2069 {
2070   \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewArea }
2071 }
2072 ,pdfviewarea / unknown .code:n =
2073 {
2074   \msg_warning:nnee { hyp } { unknown-choice+empty }
2075   { pdfviewarea }
2076   { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2077   { \exp_not:n {#1} }
2078 }
2079 ,pdfviewclip .choices:nn =
2080 { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2081 {
2082   \pdf_version_compare:NnTF < {2.0}
2083   {
2084     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }

```

```

2085         { ViewClip } { /#1 }
2086     }
2087     {
2088         \msg_warning:nnee
2089         {hyp}
2090         {ignore-deprecated-or-unknown-option-in-pdf-version}
2091         {pdfviewclip}
2092         {\pdf_version:}
2093     }
2094 }%
2095 ,pdfviewclip / .code:n =
2096 {
2097     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { ViewClip }
2098 }
2099 ,pdfviewclip / unknown .code:n =
2100 {
2101     \msg_warning:nneee { hyp } { unknown-choice+empty }
2102     { pdfviewclip }
2103     { MediaBox, CropBox, BleedBox, TrimBox, ArtBox }
2104     { \exp_not:n {#1} }
2105 }
2106 ,pdfwindowui .choice:
2107 ,pdfwindowui / true .code:n =
2108 {
2109     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } { HideWindowUI }
2110 }
2111 ,pdfwindowui / false .code:n =
2112 {
2113     \pdfmanagement_add:nnn {Catalog / ViewerPreferences }
2114     { HideWindowUI } { true }
2115 }
2116 ,pdfwindowui / .code:n =
2117 {
2118     \pdfmanagement_remove:nn {Catalog / ViewerPreferences } {HideWindowUI }
2119 }
2120 ,pdfwindowui / unknown .code:n =
2121 {
2122     \msg_warning:nnee { hyp } { no-bool }
2123     { pdfwindowui }
2124     { \exp_not:n {#1} }
2125 }
2126 ,pdfwindowui .default:n = true
2127 }

```

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

```

2128 \keys_define:nn { hyp }
2129 {
2130     ,pdfview .code:n =
2131     {
2132         \seq_set_split:Nnn \l__hyp_tmpa_seq {~}{#1}
2133         \str_case_e:nnF { \str_lowercase:f{ \seq_item:Nn \l__hyp_tmpa_seq {1} } }
2134         {
2135             { xyz }
2136         }

```



```

2137 \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } > { 1 }
2138 {
2139   \seq_get_right:NN \l__hyp_tmpa_seq \l__hyp_tmpa_tl
2140   \tl_if_eq:NnTF \l__hyp_tmpa_tl {null}
2141   {
2142     \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2143   }
2144   {
2145     \tl_set:Ne \l__hyp_dest_pdfview_tl
2146     {
2147       \fp_eval:n { \l__hyp_tmpa_tl * 100 }
2148     }
2149   }
2150 }
2151 {
2152   \tl_set:Nn \l__hyp_dest_pdfview_tl {xyz}
2153 }
2154 }
2155 { fit } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fit} }
2156 { fitb } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitb} }
2157 { fitbh } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbh} }
2158 { fitbv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitbv} }
2159 { fith } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fith} }
2160 { fitv } { \tl_set:Nn \l__hyp_dest_pdfview_tl {fitv} }
2161 { fitr }
2162 {
2163   \int_compare:nNnTF {\seq_count:N \l__hyp_tmpa_seq } = {1}
2164   {
2165     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitr}
2166   }
2167   {
2168     %ensure 4 values ...
2169     \tl_set:Nn \l__hyp_dest_pdfview_tl {fitrbox}
2170     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2171     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2172     \seq_put_right:Nn \l__hyp_tmpa_seq {0}
2173     \hbox_set_to_wd:Nnn \l__hyp_dest_box
2174     {
2175       \fp_eval:n
2176       {
2177         round
2178         (
2179           abs
2180           (
2181             \seq_item:Nn\l__hyp_tmpa_seq{4}
2182             -
2183             (\seq_item:Nn\l__hyp_tmpa_seq{2})
2184           ),
2185           3
2186         )
2187       }bp
2188     }{ }
2189     \box_set_dp:Nn \l__hyp_dest_box
2190     {

```

```

2191         \fp_eval:n
2192         {
2193             round(0 - (\seq_item:Nn\l__hyp_tmpa_seq{3}),3)
2194         }bp
2195     }
2196     \box_set_ht:Nn \l__hyp_dest_box
2197     {
2198         \seq_item:Nn\l__hyp_tmpa_seq{5}bp
2199     }
2200 }
2201 }
2202 }
2203 {
2204     \msg_warning:nnnn {hyp}{invalid-destination-value}{#1}{pdfview}
2205     \tl_set:Nn \l__hyp_dest_pdfview_tl {fit}
2206 }
2207 }
2208 ,pdfview .initial:n = {xyz}
2209 }

```

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.

```

2210 \keys_define:nn { hyp }
2211 {
2212     ,pdflang .code:n =
2213     {
2214         \tl_if_empty:nF { #1 }
2215         {
2216             \pdfmanagement_add:nne {Catalog} { Lang } { (#1) }
2217             \AddToDocumentProperties[document]{lang}{#1}
2218         }
2219     }
2220 }

```

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*) `\regex_new:N\l__hyp_optlang_regex`

`pdfkeywords` (*setup key*) `\regex_set:Nn\l__hyp_optlang_regex {\A[([A-Za-z\-\-]+)\](.*)}`

```

2223 \cs_generate_variant:Nn\clist_item:nn{on}
2224 \cs_new_protected:Npn \__hyp_setup_info_key:nn #1 #2
2225 {
2226     \keys_define:nn { hyp }
2227     {
2228         pdf#1 .code:n =
2229         {

```

```

2230 \tl_set:Ne\l__hyp_tmpa_tl {\text_expand:n{##1}}
2231 \__hyp_store_metadata:no {pdf#1}{\l__hyp_tmpa_tl}
2232 \tl_if_empty:NTF \l__hyp_tmpa_tl
2233 {
2234   \str_case:nnF { #1 }
2235   {
2236     {creator}
2237     {
2238       \msg_info:nnn { hyp }{ empty-info-value } { pdfcreator }
2239       \pdfmanagement_add:nne {Info}{Creator}{()}
2240     }
2241     {producer}
2242     {
2243       \msg_info:nnn { hyp }{ empty-info-value } { pdfproducer }
2244       \pdfmanagement_add:nne {Info}{Producer}{()}
2245     }
2246   }
2247   {
2248     \pdfmanagement_remove:nn {Info}{#2}
2249   }
2250 }
2251 {
2252   \tl_set:Ne\l__hyp_tmpb_tl {\clist_item:on{\l__hyp_tmpa_tl}{1}}
2253   \exp_args:NNV
2254   \regex_extract_once:NnN \l__hyp_optlang_regex \l__hyp_tmpb_tl\l__hyp_tmpa_s
2255   \seq_if_empty:NTF\l__hyp_tmpa_seq
2256   {
2257     \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2258   }
2259   {
2260     \__hyp_text_pdfstring_info:eN {\seq_item:Nn \l__hyp_tmpa_seq{3}}\l__hyp_t
2261   }
2262   \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2263   {
2264     \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}
2265   }
2266 }
2267 }
2268 }
2269 \keys_define:nn { hyp / info }
2270 {
2271   #2 .code:n =
2272   {
2273     \tl_set:Ne\l__hyp_tmpa_tl {\text_expand:n{##1}}
2274     \__hyp_store_metadata:eo {pdf\str_lowercase:n{##1}}{\l__hyp_tmpa_tl}
2275     \tl_if_blank:nTF {##1}
2276     {
2277       \pdfmanagement_remove:nn {Info}{#2}
2278     }
2279     {
2280       \__hyp_text_pdfstring_info:oN {\l__hyp_tmpa_tl}\l__hyp_tmpa_str
2281       \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2282       {
2283         \pdfmanagement_add:nne {Info}{#2}{\l__hyp_tmpa_str}

```

```

2284         }
2285     }
2286 }
2287 ,unknown .code:n =
2288 {
2289     \__hyp_text_pdfstring_info:eN {##1}\l__hyp_tmpa_str
2290     \str_if_eq:VnF\l__hyp_tmpa_str{<FEFF>}
2291     {
2292         \exp_args:Nno
2293         \pdfmanagement_add:nne {Info}
2294         { \l_keys_key_str } {\l__hyp_tmpa_str}
2295     }
2296 }
2297 }
2298 }
2299 \__hyp_setup_info_key:nn {author} {Author}
2300 \__hyp_setup_info_key:nn {title} {Title}
2301 \__hyp_setup_info_key:nn {producer} {Producer}
2302 \__hyp_setup_info_key:nn {creator} {Creator}
2303 % ignored key: addtopdfcreator
2304 \__hyp_setup_info_key:nn {subject} {Subject}
2305 \__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)
2306 \cs_new_protected:Npn \__hyp_setup_info_date_key:nn #1 #2
2307 {
2308     \keys_define:nn { hyp }
2309     {
2310         pdf#1 .code:n =
2311         {
2312             \tl_if_blank:nTF {##1}
2313             {
2314                 \pdfmanagement_remove:nn {Info}{#2}
2315             }
2316             {
2317                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2318             }
2319             \__hyp_store_metadata:nn {pdf#1}{##1}
2320             \AddToDocumentProperties[document]{#1}{##1}
2321         }
2322     }
2323     \keys_define:nn { hyp / info }
2324     {
2325         #2 .code:n =
2326         {
2327             \tl_if_blank:nTF {##1}
2328             {
2329                 \pdfmanagement_remove:nn {Info}{#2}
2330             }
2331             {
2332                 \pdfmanagement_add:nne {Info}{#2}{(##1)}
2333             }
2334             \exp_args:Ne \__hyp_store_metadata:nn {pdf\str_lowercase:n{#1}}{##1}

```

```

2335     }
2336   }
2337 }
2338
2339 \__hyp_setup_info_date_key:nn {creationdate} {CreationDate}
2340 \__hyp_setup_info_date_key:nn {moddate} {ModDate}
2341 \keys_define:nn { hyp }
2342 {
2343   pdfmetadate .code:n = { \__hyp_store_metadata:nn {pdfmetadate}{#1} }
2344 }

```

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

```

2345 \keys_define:nn { hyp }
2346 {
2347   ,pdftrapped .code:n =
2348   {
2349     \exp_args:Nne
2350     \keys_set:nn { hyp } { _pdftrapped = \str_uppercase:n { #1 } }
2351   }
2352   ,_pdftrapped .choices:nn = {TRUE,FALSE,UNKNOWN}
2353   {
2354     \pdfmanagement_add:nne {Info}{Trapped}
2355     {/
2356       \str_uppercase:f { \str_head:n { #1 } }
2357       \str_lowercase:f { \str_tail:n { #1 } }
2358     }
2359     \__hyp_store_metadata:ne {pdftrapped}
2360     {
2361       \str_uppercase:f { \str_head:n { #1 } }
2362       \str_lowercase:f { \str_tail:n { #1 } }
2363     }
2364   }
2365   ,_pdftrapped / unknown .code:n =
2366   {
2367     \msg_warning:nnee { hyp } { unknown-choice }
2368     { pdftrapped }
2369     { true~(case-insensitive), false~(case-insensitive), unknown~(case-insensitive) }
2370     { \exp_not:n {#1} }
2371   }
2372 }

```

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

```

2373 \keys_define:nn { hyp }
2374 {
2375   pdfinfo .code:n =
2376   {
2377     \keys_set:nn { hyp / info } { #1 }
2378   }
2379 }

```

Now we set some default values

```

2380 \keys_set:nn { hyp } {pdfcreator = LaTeX~with~hyperref}
2381 \keys_set:nn { hyp } {pdfauthor = }
2382 \keys_set:nn { hyp } {pdftitle = }
2383 \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:

```
2384 \clist_map_inline:nn
2385 {
2386   ,pdfcopyright
2387   ,pdftype
2388   ,pdflicenseurl
2389   ,pdfauthortitle
2390   ,pdfcaptionwriter
2391   ,pdfmetalang
2392   ,pdfsource
2393   ,pdfdocumentid
2394   ,pdfinstanceid
2395   ,pdfversionid
2396   ,pdfrendition
2397   ,pdfpublication
2398   ,pdfpubtype
2399   ,pdfbytes
2400   ,pdfnumpages
2401   ,pdfissn
2402   ,pdfeissn
2403   ,pdfisbn
2404   ,pdfbookedition
2405   ,pdfpublisher
2406   ,pdfvolumenum
2407   ,pdfissuenum
2408   ,pdfpagerange
2409   ,pdfdoi
2410   ,pdfurl
2411   ,pdfidentifier
2412   ,pdfsubtitle
2413   ,pdfpubstatus
2414   ,pdfcontactaddress
2415   ,pdfcontactcity
2416   ,pdfcontactregion
2417   ,pdfcontactpostcode
2418   ,pdfcontactcountry
2419   ,pdfcontactphone
2420   ,pdfcontactemail
2421   ,pdfcontacturl
2422   ,pdfdate
```

these are not hyperxmp but new keys

```
2423   ,pdftdmreservation
2424   ,pdftdmpolicy
2425 }
2426 {
2427   \keys_define:nn { hyp }
2428   {
2429     #1 .code:n= { \__hyp_store_metadata:nn {#1}{##1}}
2430   }
2431 }
2432
```

12.5 Transitions

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

```

2433 \keys_define:nn { hyp }
2434 {
2435   pdfpageduration .code:n =
2436   {
2437     \tl_if_blank:nTF { #1 }
2438     {
2439       \pdfmanagement_remove:nn {Page}{Dur}
2440     }
2441     {
2442       \pdfmanagement_add:nnn {Page}{Dur}{#1}
2443     }
2444   }
2445 }

```

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

```

2446 \keys_define:nn { hyp }
2447 {
2448   pdfpagetransition .code:n =
2449   {
2450     \tl_if_blank:nTF {#1}
2451     {
2452       \pdfmanagement_remove:nn {Page}{Trans}
2453     }

```

```

2454     {
2455         \group_begin:
2456         \keys_set:nn { hyp / trans }{style=R,#1}
2457         \pdf_object_unnamed_write:ne { dict }
2458         {
2459             \pdfdict_use:n {l__hyp_page/Trans}
2460         }
2461         \pdfmanagement_add:nne {Page}{Trans}{\pdf_object_ref_last:}
2462         \group_end:
2463     }
2464 }
2465 }
2466 \keys_define:nn { hyp / trans }
2467 {
2468     ,style .choices:nn =
2469     {Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade}
2470     { \pdfdict_put:nnn {l__hyp_page/Trans}{ S }{/#1} }
2471     ,style / unknown .code:n =
2472     {
2473         \msg_warning:nnee { hyp } { unknown-choice }
2474         { trans / style }
2475         { Split,Blinds,Box,Wipe,Dissolve,Glitter,R,Fly,Push,Cover,Uncover,Fade }
2476         { \exp_not:n {#1} }
2477     }
2478     ,duration .code:n =
2479     {
2480         \pdfdict_put:nnn {l__hyp_page/Trans}{ D }{/#1}
2481     }
2482     ,direction .choices:nn =
2483     {H,V}
2484     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Dm }{/#1} }
2485     ,direction .choices:nn =
2486     {0,90,180,270,315}
2487     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ #1 } }
2488     ,direction / None .code:n =
2489     { \pdfdict_put:nnn {l__hyp_page/Trans}{ Di }{ /None } }
2490     ,direction / unknown .code:n =
2491     {
2492         \msg_warning:nnee { hyp } { unknown-choice }
2493         { trans / direction }
2494         {
2495             H~(horizontal,~only~Split,~Blinds),
2496             V~(vertical,~only~Split,~Blinds),
2497             0~(left~to~right,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2498             90~(bottom~to~top,~only~Wipe),
2499             180~(right~to~left,~only~Wipe),
2500             270~(top~to~bottom,~only~Wipe,~Glitter,~Fly,~Cover,~Uncover,~Push),
2501             315~(top~left~to~bottom,~only~Glitter),
2502             None~(only~Fly)
2503         }
2504         { \exp_not:n {#1} }
2505     }
2506     ,motion .choices:nn =
2507     {I,0}

```



```

2508 { \pdfdict_put:nnn {l__hyp_page/Trans}{ M }{/#1} }
2509 ,motion / unknown .code:n =
2510 {
2511   \msg_warning:nneee { hyp } { unknown-choice }
2512   { trans / motion }
2513   { I~(inwards) , O~(outwards) }
2514   { \exp_not:n {#1} }
2515 }
2516 ,scale .code:n =
2517 { \pdfdict_put:nnn { l__hyp_page/Trans }{ SS }{ #1 } }
2518 ,opaque .choices:nn = {true,false}
2519 { \pdfdict_put:nnn { l__hyp_page/Trans }{ B } { #1 } }
2520 ,opaque / unknown .code:n =
2521 {
2522   \msg_warning:nneee { hyp } { unknown-choice }
2523   { trans / B }
2524   { true~(opaque~back,~only~Fly), false~(opaque~back,~only~Fly) }
2525   { \exp_not:n {#1} }
2526 }
2527 % try to set unknown keys as style
2528 ,unknown .code:n =
2529 {
2530   % warning ...
2531   \exp_args:Nne\keys_set:nn {hyp/trans}{ style=\l_keys_key_str }
2532 }
2533 }

```

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

```

2534 \keys_set_known:nv{ hyp }{opt@hyperref.sty}
    Unfinished, from here on @@ notation is disabled Form field code
2535 <@@=)
2536 \NewDocumentCommand \MakeFieldObject { m m }
2537 {
2538   \pdfxform_new:nnn { #2 }{ } { #1 }
2539 }
2540
2541
2542 \prop_new:N \g__hyp_AcroForm_CoFields_prop
2543 \prop_new:N \g__hyp_AcroForm_Fields_prop
2544
2545 \let\HyField@afields\@empty
2546 \let\HyField@cofields\@empty
2547 \def\HyField@AfterAuxOpen{\Hy@AtBeginDocument}%
2548
2549 % the value doesn't matter, but with a prop we avoid duplicates and it is
2550 % clearly faster than removing them from a sequence
2551 \def\HyField@AuxAddToFields#1
2552 {
2553   \prop_gput:Nnn \g__hyp_AcroForm_Fields_prop {#1}{F}
2554 }%
2555
2556 %fields with empty key get a value too -- lets hope that
2557 %this give the expected behaviour
2558 \def\HyField@AuxAddToCoFields #1 #2

```

```

2559 {
2560   \prop_gput:Nnn \g__hyp_AcroForm_CoFields_prop {a#1}{#2}
2561 }
2562
2563 \Hy@AtBeginDocument
2564 {
2565   \if@filesw
2566     \immediate\write\@mainaux{%
2567       \string\providecommand\string\HyField@AuxAddToFields[1]{}%
2568     }%
2569     \immediate\write\@mainaux{%
2570       \string\providecommand\string\HyField@AuxAddToCoFields[2]{}%
2571     }%
2572   \fi
2573   \let\HyField@AfterAuxOpen\@firstofone
2574 }%
2575
2576 \def\HyField@AddToFields
2577 {
2578   \exp_args:Nn\HyField@@AddToFields
2579   {
2580     \pdfannot_box_ref_last:
2581   }
2582   \ifx\Fld@calculate@code\@empty
2583   \else
2584     \begingroup
2585     \Hy@safe@activestrue
2586     \edef\Hy@temp{%
2587       \endgroup
2588       \if@filesw
2589         \write\@mainaux
2590         {
2591           \string\HyField@AuxAddToCoFields
2592           {
2593             \Fld@calculate@sortkey
2594           }
2595           {
2596             \pdfannot_box_ref_last:
2597           }
2598         }
2599       \fi
2600     }%
2601     \Hy@temp
2602   \fi
2603 }%
2604
2605 \def\HyField@@AddToFields#1{
2606   \HyField@AfterAuxOpen{%
2607     \if@filesw
2608       \write\@mainaux{%
2609         \string\HyField@AuxAddToFields{#1}%
2610       }%
2611     \fi
2612   }%

```

```

2613 }%
2614
2615 \ExplSyntaxOff
2616 \ExplSyntaxOn
2617
2618 \def\@Form[#1]
2619 {
2620   \kvsetkeys{Form}{#1}
2621   \pdf@ifdraftmode{}
2622   {
2623     \Hy@FormObjects
2624     \prop_map_inline:Nn \g__hyp_AcroForm_Fields_prop
2625     {
2626       \pdfmanagement_add:nne { Catalog / AcroForm } { Fields }{##1}
2627       %\pdfmanagement_show:n { Catalog / AcroForm }
2628     }
2629     \prop_if_empty:NF \g__hyp_AcroForm_CoFields_prop
2630     {
2631       \prop_map_inline:Nn \g__hyp_AcroForm_CoFields_prop
2632       {
2633         \seq_put_right:Nn \l__hyp_tmpa_seq {##1}
2634       }
2635       \seq_sort:Nn \l__hyp_tmpa_seq
2636       {
2637         \str_compare:nNnTF {##1} > {##2}
2638         { \sort_return_swapped: }
2639         { \sort_return_same: }
2640       }
2641       \seq_map_inline:Nn \l__hyp_tmpa_seq
2642       {
2643         \pdfmanagement_add:nne { Catalog / AcroForm }
2644         { CO }
2645         {
2646           \prop_item:Nn \g__hyp_AcroForm_CoFields_prop {##1}
2647         }
2648       }
2649     }
2650     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2651     {ZaDb} {\pdf_object_ref:n {__hyp/Font/ZaDb} }
2652     \pdfmanagement_add:nne {Catalog / AcroForm/DR/Font }
2653     {Helv} {\pdf_object_ref:n {__hyp/Font/Helv} }
2654     \pdfmanagement_add:nne {Catalog /AcroForm}
2655     {DA}{(/Helv~10~Tf~0~g)}
2656     \pdfmeta_standard_verify:nTF {form_no_NeedAppearance}
2657     {
2658       \legacy_if:nT { HyField@NeedAppearances }
2659       {
2660         \pdfmanagement_add:nnn {Catalog / AcroForm }{NeedAppearances}{true}
2661       }
2662     }
2663     {
2664       \pdfmanagement_remove:nn {Catalog / AcroForm }{NeedAppearances}
2665     }
2666   }

```

```

2667 }
2668 \ExplSyntaxOff
2669 \let\@endForm\@empty
2670 \let\HyAnn@AbsPageLabel\@empty
2671 \let\Fld@pageobjref\@empty
2672
2673 \ExplSyntaxOn
2674 \newcount\HyAnn@Count
2675 \HyAnn@Count=\z@
2676 \def\HyAnn@AbsPageLabel
2677 {
2678   \global\advance\HyAnn@Count by\@ne
2679   \__hyp_property_record:ee {HyAnn@\the\HyAnn@Count}{abspage}
2680   \property_ref_undefined_warn:ee {HyAnn@\the\HyAnn@Count}{abspage}
2681 }%
2682 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {T}
2683 \def\Fld@pageobjref
2684 {
2685   \property_if_recorded:eeT {HyAnn@\the\HyAnn@Count}{abspage}
2686   {
2687     /P~\pdf_pageobject_ref:e
2688     {
2689       \property_ref:ee{HyAnn@\the\HyAnn@Count}{abspage}
2690     }
2691   }
2692 }
2693 %% check if the attr should be set through
2694 %% hooks.
2695 %% check if options are missing.
2696 \def\@TextField[#1]#2{% parameters, label
2697   \def\Fld@name{#2}%
2698   \let\Fld@default\@empty
2699   \let\Fld@value\@empty
2700   \def\Fld@width{\DefaultWidthofText}%
2701   \def\Fld@height{%
2702     \ifFld@multiline
2703       \DefaultHeightofTextMultiline
2704     \else
2705       \DefaultHeightofText
2706     \fi
2707 }%
2708 \begingroup
2709   \expandafter\HyField@SetKeys\expandafter{%
2710     \DefaultOptionsofText,#1%
2711   }%
2712   \PDFForm@Name
2713   \HyField@FlagsText
2714   \ifFld@hidden\def\Fld@width{1sp}\fi
2715   \ifx\Fld@value\@empty\def\Fld@value{\Fld@default}\fi
2716   \LayoutTextField{#2}{%
2717     \leavevmode
2718     \HyAnn@AbsPageLabel
2719     \Hy@escapeform\PDFForm@Text
2720     \pdfannot_box:nnnn

```

```

2721         {\Fld@width}
2722         {\Fld@height}
2723         {Opt} %is this correct?
2724         {\PDFForm@Text}
2725         \MakeTextField{\Fld@width}{\Fld@height}
2726         \HyField@AddToFields
2727     }%
2728 \endgroup
2729 }
2730 \providecommand\@curropt{}
2731 \def\@ChoiceMenu[#1]#2#3{% parameters, label, choices
2732     \def\Fld@name{#2}
2733     \let\Fld@default\relax
2734     \let\Fld@value\relax
2735     \def\Fld@width{\DefaultWidthofChoiceMenu}
2736     \def\Fld@height{\DefaultHeightofChoiceMenu}
2737     \begingroup
2738         \Fld@menulength=0 %
2739         \@tempdima\z@
2740         \clist_map_variable:nNn { #3 } \@curropt
2741         %\@for\@curropt:=#3\do
2742         {%
2743             \expandafter\Fld@checkequals\@curropt==\%
2744             \Hy@StepCount\Fld@menulength
2745             \settowidth{\@tempdimb}{\@currDisplay}%
2746             \ifdim\@tempdimb>\@tempdima\@tempdima\@tempdimb\fi
2747         }%
2748         \advance\@tempdima by~15\p@
2749         \begingroup
2750             \HyField@SetKeys{#1}
2751         \edef\x{\endgroup
2752             \noexpand\expandafter
2753             \noexpand\HyField@SetKeys
2754             \noexpand\expandafter{%
2755                 \expandafter\noexpand\curname DefaultOptionsof%
2756                 \ifFld@radio
2757                     Radio%
2758                 \else
2759                     \ifFld@combo
2760                         \ifFld@popdown
2761                             PopdownBox%
2762                         \else
2763                             ComboBox%
2764                         \fi
2765                     \else
2766                         ListBox%
2767                     \fi
2768                 \fi
2769             \endcurname
2770         }%
2771     }\x
2772     \HyField@SetKeys{#1}%
2773     \PDFForm@Name
2774     \ifFld@hidden\def\Fld@width{1sp}\fi

```

```

2775 \ifx\Fld@value\relax
2776 \let\Fld@value\Fld@default
2777 \fi
2778 \LayoutChoiceField{#2}{%
2779 \ifFld@radio
2780 \HyField@FlagsRadioButton
2781 \@@Radio{#3}%
2782 \else
2783 \beginingroup
2784 \HyField@FlagsChoice
2785 \ifdim\Fld@width<\@tempdima
2786 \ifdim\@tempdima<1cm\@tempdima1cm\fi
2787 \edef\Fld@width{\the\@tempdima}%
2788 \fi
2789 \ifFld@combo
2790 \else
2791 \@tempdima=\the\Fld@menulength\Fld@charsize
2792 \advance\@tempdima by~\Fld@borderwidth bp %
2793 \advance\@tempdima by~\Fld@borderwidth bp %
2794 \edef\Fld@height{\the\@tempdima}%
2795 \fi
2796 \@@ListBox{#3}%
2797 \endgroup
2798 \fi
2799 }%
2800 \endgroup
2801 }
2802
2803 \def\@@Radio#1{%
2804 \Fld@listcount=0-%
2805 \EdefEscapeName\Fld@default{\Fld@default}%
2806 \clist_map_variable:nNn { #1 } \@curropt
2807 {%
2808 \expandafter\Fld@checkequals\@curropt==\%
2809 \EdefEscapeName\@currValue{\@currValue}%
2810 \Hy@StepCount\Fld@listcount
2811 \@currDisplay\space
2812 \leavevmode
2813 \HyAnn@AbsPageLabel
2814 \Hy@escapeform\PDFForm@Radio
2815 \pdfxform_if_exist:nF { __hyp_xform_Ding }
2816 {
2817 \pdfxform_new:nnn { __hyp_xform_Ding } {}
2818 {
2819 \group_begin:
2820 \fontfamily{pzd}
2821 \fontencoding{U}
2822 \fontseries{m}
2823 \fontshape{n}
2824 \selectfont
2825 \char123
2826 \group_end:
2827 }
2828 }

```

```

2829 \pdfannot_box:nnne
2830   {\Fld@width}
2831   {\Fld@height}
2832   {Opt} %is this correct?
2833   {
2834     \PDFForm@Radio
2835     /AP
2836     <<
2837     /N
2838     <<
2839     /\@currValue\c_space_tl \pdfxform_ref:n {__hyp_xform_Ding}
2840     %/Off \c_space_tl \pdfxform_ref:n {__hyp_xform_DingOff} %hm
2841     >>
2842     >>
2843   }
2844   {\fbox{ \MakeRadioField{\Fld@width}{\Fld@height}} }
2845   \int_compare:nNnT { \Fld@listcount} = { 1 }
2846   { \HyField@AddToFields }
2847   \c_space_tl % deliberate space between radio buttons
2848             % to do: --> should be configurable
2849   }%
2850 }
2851
2852 \newcount\Fld@listcount
2853 \def\@@ListBox#1
2854 {
2855   \HyField@PDFChoices{#1}
2856   \mode_leave_vertical:
2857   \HyAnn@AbsPageLabel
2858   \Hy@escapeform\PDFForm@List
2859   \pdfannot_box:nnnn
2860   {\Fld@width}{\Fld@height}{Opt}
2861   {\PDFForm@List}
2862   \MakeChoiceField{\Fld@width}{\Fld@height}
2863   \HyField@AddToFields
2864 }
2865
2866
2867 \def\@PushButton[#1]#2{% parameters, label
2868   \def\Fld@name{#2}%
2869   \group_begin:
2870     \exp_args:No\HyField@SetKeys
2871     {
2872       \DefaultOptionsofPushButton,#1
2873     }
2874     \PDFForm@Name
2875     \pdfmeta_standard_verify:nnTF {annot_action_A}{JavaScript}
2876     {
2877       \HyField@FlagsPushButton
2878       \legacy_if:nT {Fld@hidden}
2879       {
2880         \def\Fld@width{1sp}
2881       }
2882       \LayoutPushButtonField

```

```

2883     {
2884         \mode_leave_vertical:
2885         \HyAnn@AbsPageLabel
2886         \Hy@escapeform\PDFForm@Push
2887         \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2888         \pdfannot_box:nnnn
2889         {\box_wd:N\l_tmpa_box}
2890         {\box_ht:N\l_tmpa_box}
2891         {\box_dp:N\l_tmpa_box} %is this correct?
2892         {\PDFForm@Push}
2893         {\box_use:N\l_tmpa_box}
2894         \HyField@AddToFields
2895     }
2896 }
2897 {
2898     \msg_error:nn { hyp }{ pdfa-no-push-button }
2899     \LayoutPushButtonField
2900     {
2901         \mode_leave_vertical:
2902         \MakeButtonField{#2}
2903     }
2904 }
2905 \group_end:
2906 }
2907
2908 \def\@Submit[#1]#2
2909 {
2910     \def\Fld@width {\DefaultWidthofSubmit}
2911     \def\Fld@height{\DefaultHeightofSubmit}
2912     \group_begin:
2913     \exp_args:No\HyField@SetKeys
2914     {
2915         \DefaultOptionsofSubmit,#1
2916     }
2917     \HyField@FlagsPushButton
2918     \HyField@FlagsSubmit
2919     \legacy_if:nT { Fld@hidden }
2920     {
2921         \def\Fld@width{1sp}
2922     }
2923     \mode_leave_vertical:
2924     \HyAnn@AbsPageLabel
2925     \Hy@escapeform\PDFForm@Submit
2926     \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2927     \pdfxform_if_exist:nF
2928     { __hyp_xform_Submit }
2929     {
2930         \pdfxform_new:nnn { __hyp_xform_Submit }{-}
2931         {
2932             \fbox{\color_select:n{yellow}\textsf{Submit}}
2933         }
2934         \pdfxform_new:nnn { __hyp_xform_SubmitP }{-}
2935         {
2936             \fbox{\color_select:n{yellow}\textsf{SubmitP}}

```



```

2937     }
2938   }
2939   \pdfannot_box:nnnn
2940     {\box_wd:N\l_tmpa_box}
2941     {\box_ht:N\l_tmpa_box}
2942     {\box_dp:N\l_tmpa_box} %is this correct?
2943     {
2944       \PDFForm@Submit
2945       /AP<<
2946         /N~\pdfxform_ref:n {__hyp_xform_Submit}~
2947         /D~\pdfxform_ref:n {__hyp_xform_SubmitP}
2948       >>
2949     }
2950     \HyField@AddToFields
2951     \box_use:N\l_tmpa_box
2952
2953   \group_end:
2954 }
2955
2956 \def\@Reset[#1]#2
2957 {
2958   \def\Fld@width {\DefaultWidthofReset}
2959   \def\Fld@height{\DefaultHeightofReset}
2960   \group_begin:
2961     \exp_args:No\HyField@SetKeys
2962     {
2963       \DefaultOptionsofReset,#1
2964     }
2965     \mode_leave_vertical:
2966     \pdfmeta_standard_verify:nnTF {annot_action_A}{ResetForm}
2967     {
2968       \HyField@FlagsPushButton
2969       \legacy_if:nT { Fld@hidden }
2970       { \def\Fld@width{1sp} }
2971       \HyAnn@AbsPageLabel
2972       \Hy@escapeform\PDFForm@Reset
2973       \hbox_set:Nn \l_tmpa_box { \MakeButtonField {#2}}
2974       \pdfannot_box:nnnn
2975         {\box_wd:N\l_tmpa_box}
2976         {\box_ht:N\l_tmpa_box}
2977         {\box_dp:N\l_tmpa_box} %is this correct?
2978         { \PDFForm@Reset }
2979       \HyField@AddToFields
2980       \box_use:N \l_tmpa_box
2981     }
2982     {
2983       \msg_error:nn { hyp }{ pdfa-no-reset-button }
2984       \MakeButtonField{#2}
2985     }
2986   \group_end:
2987 }
2988
2989 \def\@CheckBox[#1]#2
2990 {% parameters, label

```

```

2991 \def\Fld@name{#2}
2992 \def\Fld@default{0}
2993 \group_begin:
2994 \def\Fld@width {\DefaultWidthofCheckBox}
2995 \def\Fld@height{\DefaultHeightofCheckBox}
2996 \exp_args:No\HyField@SetKeys
2997 {
2998   \DefaultOptionsofCheckBox,#1
2999 }
3000 \PDFForm@Name
3001 \HyField@FlagsCheckBox
3002 \legacy_if:nT { Fld@hidden }
3003 {
3004   \def\Fld@width{1sp}
3005 }
3006 \LayoutCheckField{#2}
3007 {
3008   \mode_leave_vertical:
3009   \HyAnn@AbsPageLabel
3010   \Hy@escapeform\PDFForm@Check
3011   \pdfxform_if_exist:nF { __hyp_xform_CheckMarkYes }
3012   {
3013     \pdfxform_new:nnn
3014     {__hyp_xform_CheckMarkYes}{-}
3015     {
3016       \group_begin:
3017       \fontfamily{pzd}
3018       \fontencoding{U}
3019       \fontseries{m}
3020       \fontshape{n}
3021       \selectfont
3022       \char51
3023       \group_end:
3024     }
3025     \pdfxform_new:nnn
3026     {__hyp_xform_CheckMarkOff}{-}
3027     {
3028       \group_begin:
3029       \fontfamily{pzd}
3030       \fontencoding{U}
3031       \fontseries{m}
3032       \fontshape{n}
3033       \selectfont
3034       \phantom{\char51} %perhaps xetex needs some small glyph ..
3035       \group_end:
3036     }
3037   }
3038   \pdfannot_box:nnnn
3039   {\Fld@width}
3040   {\Fld@height}
3041   {Opt} %is this correct?
3042   {\PDFForm@Check}
3043   \HyField@AddToFields %check if this works with xelatex ...
3044 }

```

```

3045 \group_end:
3046 }
3047 %hm. Should a luatex driver use type1 fonts in fields????
3048 \def\Hy@FormObjects
3049 {
3050 \pdf_object_new:n {__hyp/Encoding/pdfdoc }
3051 \pdf_object_new:n {__hyp/Font/ZaDb }
3052 \pdf_object_new:n {__hyp/Font/Helv }
3053 \pdf_object_write:nne {__hyp/Encoding/pdfdoc } { dict }
3054 {
3055 /Type/Encoding
3056 /Differences[
3057 24/breve/caron/circumflex/dotaccent/hungarumlaut/ogonek
3058 /ring/tilde
3059 \c_space_tl
3060 39/quotesingle
3061 \c_space_tl
3062 96/grave %
3063 \iow_newline:
3064 128/bullet/dagger/daggerdbl/ellipsis/emdash/endash/florin
3065 /fraction/guilsinglleft/guilsinglright/minus/perthousand
3066 /quotedblbase/quotedblleft/quotedblright/quoteleft
3067 /quoteright/quotesinglbase/trademark/fi/fl/Lslash/OE
3068 /Scaron/Ydieresis/Zcaron/dotlessi/lslash/oe/scaron/zcaron
3069 \iow_newline:
3070 164/currency
3071 \c_space_tl
3072 166/brokenbar
3073 \c_space_tl
3074 168/dieresis/copyright/ordfeminine
3075 \c_space_tl
3076 172/logicalnot/.notdef/registered/macron/degree/plusminus
3077 /twosuperior/threesuperior/acute/mu
3078 \c_space_tl
3079 183/periodcentered/cedilla/onesuperior/ordmasculine
3080 \c_space_tl
3081 188/onequarter/onehalf/threequarters
3082 \iow_newline:
3083 192/Agrave/Aacute/Acircumflex/Atilde/Adieresis/Aring/AE
3084 /Ccedilla/Egrave/Eacute/Ecircumflex/Edieresis/Igrave
3085 /Iacute/Icircumflex/Idieresis/Eth/Ntilde/Ograve/Oacute
3086 /Ocircumflex/Otilde/Odieresis/multiply/Oslash/Ugrave
3087 /Uacute/Ucircumflex/Udieresis/Yacute/Thorn/germandbls
3088 /agrave/aacute/acircumflex/atilde/adieresis/aring/ae
3089 /ccedilla/egrave/eacute/ecircumflex/edieresis/igrave
3090 /iacute/icircumflex/idieresis/eth/ntilde/ograve/oacute
3091 /ocircumflex/otilde/odieresis/divide/oslash/ugrave
3092 /uacute/ucircumflex/udieresis/yacute/thorn/ydieresis
3093 ]
3094 }
3095 \pdf_object_write:nnn {__hyp/Font/ZaDb } { dict }
3096 {
3097 /Type/Font
3098 /Subtype/Type1

```

```

3099     /Name/ZaDb
3100     /BaseFont/ZapfDingbats
3101   }
3102   \pdf_object_write:nne {__hyp/Font/Helv } { dict }
3103   {
3104     /Type/Font
3105     /Subtype/Type1
3106     /Name/Helv
3107     /BaseFont/Helvetica
3108     /Encoding~\pdf_object_ref:n { __hyp/Encoding/pdfdoc }
3109   }
3110   \global\let\Hy@FormObjects\relax
3111   }
3112   \ExplSyntaxOff
3113   \providecommand*{\Fld@pageobjref}{}
3114   \ifcsname pdf@escapestring\endcsname
3115     \def\Hy@escapeform#1{%
3116       \ifHy@pdfescapeform
3117         \let\Hy@escapestring\pdfescapestring
3118       \else
3119         \let\Hy@escapestring\@firstofone
3120       \fi
3121     }%
3122     \Hy@escapeform{}%
3123   \else
3124     \let\Hy@escapestring\@firstofone
3125     \def\Hy@escapeform#1{%
3126       \ifHy@pdfescapeform
3127         \def\Hy@escapestring##1{%
3128           \noexpand\Hy@escapestring{\noexpand##1}%
3129         }%
3130         \edef\Hy@temp{#1}%
3131         \expandafter\Hy@@escapeform\Hy@temp\Hy@escapestring{}\@nil
3132         \def\Hy@escapestring##1{%
3133           \@ifundefined{Hy@esc@\string##1}{%
3134             ##1%
3135             \ThisShouldNotHappen
3136           }{%
3137             \csname Hy@esc@\string##1\endcsname
3138           }%
3139         }%
3140       \else
3141         \let\Hy@escapestring\@firstofone
3142       \fi
3143     }%
3144     \def\Hy@@escapeform#1\Hy@escapestring#2#3\@nil{%
3145       \ifx\#3\%
3146         \else
3147           \expandafter
3148           \Hy@pstringdef\csname Hy@esc@\string#2\endcsname{#2}% probably string-hex
3149           \Hy@ReturnAfterFi{%
3150             \Hy@@escapeform#3\@nil
3151           }%
3152         \fi

```

```

3153 }%
3154 \fi
3155 \def\PDFForm@Name{%
3156   \PDFForm@@Name\Fld@name
3157   \ifx\Fld@altname\relax
3158   \else
3159     \PDFForm@@Name\Fld@altname
3160   \fi
3161   \ifx\Fld@mappingname\relax
3162   \else
3163     \PDFForm@@Name\Fld@mappingname
3164   \fi
3165 }
3166 \def\PDFForm@@Name#1{%
3167   \begingroup
3168     \ifnum\Hy@pdfversion<5 % implementation note 117, PDF spec 1.7
3169     \ifHy@unicode
3170       \Hy@unicodedefalse
3171     \fi
3172     \fi
3173     \pdfstringdef\Hy@gtemp#1%
3174   \endgroup
3175   \let#1\Hy@gtemp
3176 }
3177 \def\Fld@X@additionalactions{%
3178   \ifx\Fld@keystroke@code\@empty
3179   \else
3180     /K<</S/JavaScript/JS(\Hy@escapestring{\Fld@keystroke@code})>>%
3181   \fi
3182   \ifx\Fld@format@code\@empty
3183   \else
3184     /F<</S/JavaScript/JS(\Hy@escapestring{\Fld@format@code})>>%
3185   \fi
3186   \ifx\Fld@validate@code\@empty
3187   \else
3188     /V<</S/JavaScript/JS(\Hy@escapestring{\Fld@validate@code})>>%
3189   \fi
3190   \ifx\Fld@calculate@code\@empty
3191   \else
3192     /C<</S/JavaScript/JS(\Hy@escapestring{\Fld@calculate@code})>>%
3193   \fi
3194   \ifx\Fld@onfocus@code\@empty
3195   \else
3196     /Fo<</S/JavaScript/JS(\Hy@escapestring{\Fld@onfocus@code})>>%
3197   \fi
3198   \ifx\Fld@onblur@code\@empty
3199   \else
3200     /Bl<</S/JavaScript/JS(\Hy@escapestring{\Fld@onblur@code})>>%
3201   \fi
3202   \ifx\Fld@onmousedown@code\@empty
3203   \else
3204     /D<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmousedown@code})>>%
3205   \fi
3206   \ifx\Fld@onmouseup@code\@empty

```

```

3207 \else
3208 /U<</S/JavaScript/JS(\Hy@escapestring{\Fld@onmouseup@code})>>%
3209 \fi
3210 \ifx\Fld@onenter@code\@empty
3211 \else
3212 /E<</S/JavaScript/JS(\Hy@escapestring{\Fld@onenter@code})>>%
3213 \fi
3214 \ifx\Fld@onexit@code\@empty
3215 \else
3216 /X<</S/JavaScript/JS(\Hy@escapestring{\Fld@onexit@code})>>%
3217 \fi
3218 }
3219 \ExplSyntaxOn
3220 \def\Fld@additionalactions
3221 {%
3222 \exp_args:N\str_if_eq:nnF {\Fld@X@additionalactions}{}
3223 {
3224 \pdfmeta_standard_verify:nT {annot_widget_no_AA}
3225 {/AA<<\Fld@X@additionalactions>>}
3226 }
3227 }
3228 \def\Fld@annotnames{%
3229 /T(\Fld@name)%
3230 \ifx\Fld@altname\relax
3231 \else
3232 /TU(\Fld@altname)%
3233 \fi
3234 \ifx\Fld@mappingname\relax
3235 \else
3236 /TM(\Fld@mappingname)%
3237 \fi
3238 }
3239
3240 \def\PDFForm@Check
3241 {
3242 /Subtype/Widget
3243 ~\Fld@annotflags
3244 ~\Fld@pageobjref
3245 ~\Fld@annotnames
3246 /FT/Btn
3247 \Fld@flags
3248 /Q~\Fld@align
3249 /BS<</W~\Fld@borderwidth /S/\Fld@borderstyle>>
3250 /AP
3251 <<
3252 /N
3253 <<
3254 /Yes~\pdfxform_ref:n{__hyp_xform_CheckMarkYes}
3255 /Off~\pdfxform_ref:n{__hyp_xform_CheckMarkOff}
3256 >>
3257 >>
3258 /MK<<
3259 \int_compare:nNnF {\Fld@rotation}={0}
3260 {

```

```

3261         /R~\Fld@rotation
3262     }
3263     \tl_if_empty:NF\Fld@bordercolor
3264     {
3265         /BC[\Fld@bordercolor]
3266     }
3267     \tl_if_empty:NF\Fld@bcolor
3268     {
3269         /BG[\Fld@bcolor]
3270     }
3271     /CA(\Hy@escapestring{\Fld@cbsymbol})%
3272 >>
3273 /DA
3274 (
3275     /ZaDb~\strip@pt\Fld@charsize\c_space_tl Tf
3276     \tl_if_empty:NF \Fld@color
3277     {
3278         \c_space_tl \Fld@color
3279     }
3280 )
3281 /H/P
3282 \legacy_if:nTF {Fld@checked}
3283 {
3284     /V/Yes /AS/Yes
3285 }
3286 {
3287     /V/Off /AS/Off
3288 }
3289 \Fld@additionalactions
3290 }
3291
3292 \def\PDFForm@Push
3293 {
3294     /Subtype/Widget
3295     ~\Fld@annotflags
3296     ~\Fld@pageobjref
3297     ~\Fld@annotnames
3298     /FT/Btn
3299     ~\Fld@flags
3300     /H/P
3301     /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3302     \bool_if:nT
3303     {
3304         !\int_compare_p:nNn {\Fld@rotation} = {0}
3305         ||
3306         \tl_if_exist_p:N \Fld@bordercolor
3307     }
3308     {
3309         /MK
3310         <<
3311             \int_compare:nNnF {\Fld@rotation} = {0}
3312             {
3313                 /R~\Fld@rotation
3314             }

```

```

3315         \tl_if_exist:NT \Fld@bordercolor
3316         {
3317             /BC[\Fld@bordercolor]
3318         }
3319     >>
3320 }
3321 /A<</S/JavaScript/JS(\Hy@escapestring{\Fld@onclick@code})>>
3322 \Fld@additionalactions
3323 }
3324
3325 \ExplSyntaxOff
3326 \def\PDFForm@List{%
3327     /Subtype/Widget%
3328     \Fld@annotflags%
3329     \Fld@pageobjref
3330     \Fld@annotnames
3331     /FT/Ch%
3332     \Fld@flags
3333     /Q \Fld@align
3334     /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3335     \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3336         \ifx\Fld@bordercolor\relax\else 1\fi
3337         \ifx\Fld@bcolor\relax \else 1\fi
3338         \space
3339     \else
3340         /MK<<%
3341         \ifnum\Fld@rotation=\z@
3342         \else
3343             /R \Fld@rotation
3344         \fi
3345         \ifx\Fld@bordercolor\relax
3346         \else
3347             /BC[\Fld@bordercolor]%
3348         \fi
3349         \ifx\Fld@bcolor\relax
3350         \else
3351             /BG[\Fld@bcolor]%
3352         \fi
3353         >>%
3354     \fi
3355     /DA(/Helv \strip@pt\Fld@charsize\space Tf%
3356         \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3357     \Fld@choices
3358     \Fld@additionalactions
3359 }
3360 \ExplSyntaxOn
3361 \def\PDFForm@Radio
3362 {
3363     /Subtype/Widget
3364     ~\Fld@annotflags
3365     ~\Fld@pageobjref
3366     ~\Fld@annotnames
3367     /FT/Btn
3368     \Fld@flags

```



```

3369 /H/P
3370 /BS<</W~\Fld@borderwidth/S/\Fld@borderstyle>>
3371 /MK<<
3372 \ifnum\Fld@rotation=\z@
3373 \else
3374 /R~\Fld@rotation
3375 \fi
3376 \ifx\Fld@bordercolor\relax
3377 \else
3378 /BC[\Fld@bordercolor]%
3379 \fi
3380 \ifx\Fld@bcolor\relax
3381 \else
3382 /BG[\Fld@bcolor]%
3383 \fi
3384 /CA(\Hy@escapestring{\Fld@radiosymbol})%
3385 >>
3386 /DA(/ZaDb~\strip@pt\Fld@charsize\space Tf%
3387 \ifx\Fld@color\@empty\else\space\Fld@color\fi)%
3388 \ifx\Fld@default\@empty
3389 /V/Off%
3390 /DV/Off%
3391 \else
3392 /V/\Fld@default
3393 /DV/\Fld@default
3394 \fi
3395 \Fld@additionalactions
3396 }
3397
3398 % Does an appearance dict make sense here?
3399 \def\PDFForm@Text
3400 {
3401 /Subtype/Widget
3402 ~\Fld@annotflags
3403 ~\Fld@pageobjref
3404 ~\Fld@annotnames
3405 /FT/Tx
3406 ~\Fld@flags
3407 /Q~\Fld@align
3408 /BS<</W~\Fld@borderwidth\c_space_t1 /S /\Fld@borderstyle>>
3409 \bool_if:nT
3410 {
3411 !\int_compare_p:nNn {\Fld@rotation} = {0}
3412 ||
3413 \tl_if_exist_p:N \Fld@bordercolor
3414 ||
3415 \tl_if_exist_p:N \Fld@bcolor
3416 }
3417 {
3418 /MK
3419 <<
3420 \int_compare:nNnF {\Fld@rotation} = {0}
3421 {
3422 /R~\Fld@rotation

```

```

3423     }
3424     \tl_if_exist:NT \Fld@bordercolor
3425     {
3426         /BC[\Fld@bordercolor]
3427     }
3428     \tl_if_exist:NT \Fld@bcolor
3429     {
3430         /BG[\Fld@bcolor]
3431     }
3432     >>
3433 }
3434 /DA
3435 (
3436     /Helv~\strip@pt\Fld@charsize\c_space_tl Tf
3437     \tl_if_empty:NF {\c_space_tl\Fld@color}
3438 )
3439 /DV(\Hy@escapestring{\Fld@default})
3440 /V(\Hy@escapestring{\Fld@value})
3441 ~\Fld@additionalactions
3442 \int_compare:nNnT { \Fld@maxlen}>{0}
3443 {
3444     /MaxLen~\Fld@maxlen
3445 }
3446 }
3447 \ExplSyntaxOff
3448
3449 \def\PDFForm@Submit{%
3450 /Subtype/Widget%
3451 \Fld@annotflags
3452 \Fld@pageobjref
3453 \Fld@annotnames
3454 /FT/Btn%
3455 \Fld@flags
3456 /H/P%
3457 /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3458 \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3459     \ifx\Fld@bordercolor\relax\else 1\fi
3460     \space
3461 \else
3462     /MK<<%
3463     \ifnum\Fld@rotation=\z@
3464     \else
3465     /R \Fld@rotation
3466     \fi
3467     \ifx\Fld@bordercolor\relax
3468     \else
3469     /BC[\Fld@bordercolor]%
3470     \fi
3471     >>%
3472 \fi
3473 /A<<%
3474 /S/SubmitForm%
3475 /F<<%
3476 /FS/URL%

```

```

3477     /F(\Hy@escapestring{\Form@action})%
3478     >>%
3479     \Fld@submitflags
3480     >>%
3481     \Fld@additionalactions
3482 }
3483 \ExplSyntaxOn
3484     \def\PDFForm@Reset{%
3485         /Subtype/Widget%
3486         \Fld@annotflags
3487         \Fld@pageobjref
3488         \Fld@annotnames
3489         /FT/Btn%
3490         \Fld@flags
3491         /H/P%
3492         /DA(/Helv~\strip@pt\Fld@charsize\space Tf~0~0~1~rg)%
3493         \ifcase0\ifnum\Fld@rotation=\z@ \else 1\fi
3494             \ifx\Fld@bordercolor\relax\else 1\fi
3495             \space
3496         \else
3497             /MK<<%
3498             \ifnum\Fld@rotation=\z@
3499             \else
3500             /R~\Fld@rotation
3501             \fi
3502             \ifx\Fld@bordercolor\relax
3503             \else
3504             /BC[\Fld@bordercolor]%
3505             \fi
3506             >>%
3507             \fi
3508             /BS<</W \Fld@borderwidth/S/\Fld@borderstyle>>%
3509             /A<</S/ResetForm>>%
3510             \Fld@additionalactions
3511         }%
3512
3513
3514 %these patterns are used in hyperref checks.
3515 %it is unclear if they are really useful and if a backend support is
3516 %needed.
3517 \str_case:VnF \c_sys_backend_str
3518 {
3519     { pdfmode }
3520     {
3521         \def\HyPat@ObjRef
3522         {
3523             [0-9]*[1-9][0-9]*~0~R
3524         }
3525     }
3526     { dvipdfmx }
3527     {
3528         \def\HyPat@ObjRef
3529         {
3530             @[\~]+

```

```

3531     }
3532   }
3533   { xdvipdfmx }
3534   {
3535     \def\HyPat@ObjRef
3536     {
3537       @[\~]+
3538     }
3539   }
3540 }
3541 { %also set in hyperref sty, so probably not needed.
3542   \def\HyPat@ObjRef/{.+}
3543 }
3544
3545
3546 \ExplSyntaxOff
3547 % UF: removed Hy@writebookmark
3548 %   \Hy@currentbookmarklevel{0}
3549 %   \Hy@numberline
3550 %   \@@writetorep
3551 %   counter{bookmark@seq@number}
3552 % removed \HyPsd@SanitizeForOutFile, not needed
3553 % removed \currentpdfbookmark, defined by bookmark,
3554 % should use \newcommand there
3555 % removed \subpdfbookmark, defined by bookmark,
3556 % should use \newcommand there
3557 % removed \belowpdfbookmark, defined by bookmark,
3558 % should use \newcommand there
3559 % removed \pdfbookmark, defined by bookmark,
3560 % \BOOKMARK
3561 % \@BOOKMARK
3562 %% \RequirePackage{rerunfilecheck}[2009/12/10]
3563 %% removed \Hy@OutlineRerunCheck, unneeded with bookmark
3564 %% removed \ReadBookmarks / unneeded with bookmark.
3565 %% removed \Hy@OutlineName
3566 %% removed \check@bm@number
3567 %% removed \calc@bm@number
3568
3569 \ifHy@implicit
3570 \else
3571   \expandafter\endinput
3572 \fi
3573 \newlength\Hy@SectionHShift
3574 \def\Hy@SectionAnchorHref#1{%
3575   \ifx\protect\@typeset@protect
3576     \Hy@@SectionAnchor{#1}%
3577   \fi
3578 }
3579 \DeclareRobustCommand*\Hy@@SectionAnchor}[1]{%
3580   \leavevmode
3581   \hbox to Opt{%
3582     \kern-\Hy@SectionHShift
3583     \Hy@raisedlink{%
3584       \hyper@anchorstart{#1}\hyper@anchorend

```

```

3585     }%
3586     \hss
3587   }%
3588 }
3589 \@ifundefined{hyper@nopatch@sectioning}
3590 {
3591   \let\H@old@ssect\@ssect
3592   \def\@ssect#1#2#3#4#5{%
3593     \Hy@MakeCurrentHrefAuto{section*}%
3594     \setlength{\Hy@SectionHShift}{#1}%
3595     \begingroup
3596       \toks@{\H@old@ssect{#1}{#2}{#3}{#4}}%
3597       \toks\tw@\expandafter{%
3598         \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3599         #5%
3600       }%
3601     \edef\x{\endgroup
3602       \the\toks@\the\toks\tw@}%
3603   }\x
3604 }
3605 \let\H@old@schapter\@schapter
3606 \def\@schapter#1{%
3607   \begingroup
3608     \let\@mkboth\@gobbletwo
3609     \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3610     \Hy@raisedlink{%
3611       \hyper@anchorstart{\@currentHref}\hyper@anchorend
3612     }%
3613   \endgroup
3614   \H@old@schapter{#1}%
3615 }
3616 \@ifundefined{chapter}{-}{%
3617   \let\Hy@org@chapter\@chapter
3618   \def\@chapter{%
3619     \def\Hy@next{%
3620       \Hy@MakeCurrentHrefAuto{\Hy@chapapp*}%
3621       \Hy@raisedlink{%
3622         \hyper@anchorstart{\@currentHref}\hyper@anchorend
3623       }%
3624     }%
3625     \ifnum\c@secnumdepth>\m@ne
3626       \@ifundefined{if@mainmatter}%
3627       \iftrue{\csname if@mainmatter\endcsname}%
3628       \let\Hy@next\relax
3629     \fi
3630     \fi
3631     \Hy@next
3632     \Hy@org@chapter
3633   }%
3634 }
3635 \let\H@old@part\@part
3636 \begingroup\expandafter\expandafter\expandafter\endgroup
3637 \expandafter\ifx\csname chapter\endcsname\relax
3638   \let\Hy@secnum@part\z@

```

```

3639 \else
3640   \let\Hy@secnum@part\m@ne
3641 \fi
3642 \def\@part{%
3643   \ifnum\Hy@secnum@part>\c@secnumdepth
3644     \phantomsection
3645   \fi
3646   \H@old@part
3647 }
3648 \let\H@old@spart\@spart
3649 \def\@spart#1{%
3650   \Hy@MakeCurrentHrefAuto{part*}%
3651   \Hy@raisedlink{%
3652     \hyper@anchorstart{\@currentHref}\hyper@anchorend
3653   }%
3654   \H@old@spart{#1}%
3655 }
3656 \let\H@old@sect\@sect
3657 \def\@sect#1#2#3#4#5#6[#7]#8{%
3658   \ifnum #2>\c@secnumdepth
3659     \expandafter\@firstoftwo
3660   \else
3661     \expandafter\@secondoftwo
3662   \fi
3663   {%
3664     \Hy@MakeCurrentHrefAuto{section*}%
3665     \setlength{\Hy@SectionHShift}{#3}%
3666     \begingroup
3667     \toks@{\H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]]}%
3668     \toks\tw@\expandafter{%
3669       \expandafter\Hy@SectionAnchorHref\expandafter{\@currentHref}%
3670       #8%
3671     }%
3672     \edef\x{\endgroup
3673       \the\toks@\the\toks\tw@}%
3674     }\x
3675   }{%
3676     \H@old@sect{#1}{#2}{#3}{#4}{#5}{#6}[[#7]][#8]%
3677   }%
3678 }
3679 }{}
3680 \expandafter\def\csname Parent-4\endcsname{}
3681 \expandafter\def\csname Parent-3\endcsname{}
3682 \expandafter\def\csname Parent-2\endcsname{}
3683 \expandafter\def\csname Parent-1\endcsname{}
3684 \expandafter\def\csname Parent0\endcsname{}
3685 \expandafter\def\csname Parent1\endcsname{}
3686 \expandafter\def\csname Parent2\endcsname{}
3687 \expandafter\def\csname Parent3\endcsname{}
3688 \expandafter\def\csname Parent4\endcsname{}
3689 %%
3690 </package>
3691 <@@=hyp>
3692 <*colorscheme>

```

```

3693 % collected from https://tex.stackexchange.com/questions/525261/better-default-
      colors-for-hyperref-links
3694 % cite color ignored, as it doesn't fit ... should be done by cite packages ?
3695 % linkcolor=
3696 %,filecolor=
3697 %,urlcolor=
3698 %,menucolor=
3699 %,runcolor=
3700 %,linkbordercolor=
3701 %,filebordercolor=
3702 %,urlbordercolor=
3703 %,menubordercolor=
3704 %,runbordercolor=
3705
3706 \prop_const_from_keyval:cn { c__hyp_colorscheme_primary-colors_prop }
3707 {
3708   linkcolor      = [rgb]{1,0,0}, %red
3709   filecolor      = [rgb]{0,1,1}, %cyan
3710   urlcolor       = [rgb]{1,0,1}, %magenta
3711   menucolor      = [rgb]{1, 0, 0}, %red
3712   runcolor       = [rgb]{0,1,1}, %cyan
3713 %-----
3714   linkbordercolor = [rgb]{1, 0 ,0 },
3715   filebordercolor = [rgb]{0, .5, .5},
3716   urlbordercolor  = [rgb]{0, 1, 1},
3717   menubordercolor = [rgb]{1, 0, 0},
3718   runbordercolor  = [rgb]{0, .7, .7}
3719 }
3720
3721 \prop_const_from_keyval:Nn \c__hyp_colorscheme_daleif_prop
3722 {
3723   linkcolor      = [rgb]{0,0.2,0.6},
3724   filecolor      = [rgb]{0.8,0,0.8},
3725   urlcolor       = [rgb]{0.8,0,0.8},
3726   menucolor      = [rgb]{0,0.2,0.6},
3727   runcolor       = [rgb]{0.8,0,0.8},
3728 %----- %-----
3729   linkbordercolor = [rgb]{0,0.2,0.6},
3730   filebordercolor = [rgb]{0.8,0,0.8},
3731   urlbordercolor  = [rgb]{0.8,0,0.8},
3732   menubordercolor = [rgb]{0,0.2,0.6},
3733   runbordercolor  = [rgb]{0.8,0,0.8}
3734 }
3735
3736 \prop_const_from_keyval:Nn \c__hyp_colorscheme_julian_prop
3737 { %two colors: intern/extern
3738   linkcolor      = [rgb]{0.79216, 0, 0.12549},
3739   filecolor      = [rgb]{0.01961, 0.44314, 0.6902},
3740   urlcolor       = [rgb]{0.01961, 0.44314, 0.6902},
3741   menucolor      = [rgb]{0.79216, 0, 0.12549 },
3742   runcolor       = [rgb]{0.01961, 0.44314, 0.6902 },
3743 %----- %-----
3744   linkbordercolor = [rgb]{0.79216, 0, 0.12549},
3745   filebordercolor = [rgb]{0.01961, 0.44314, 0.6902},

```

```

3746     urlbordercolor = [rgb]{0.01961, 0.44314, 0.6902},
3747     menubordercolor = [rgb]{0.79216, 0, 0.12549 },
3748     runbordercolor = [rgb]{0.01961, 0.44314, 0.6902 }
3749 }
3750
3751 \prop_const_from_keyval:Nn \c__hyp_colorscheme_tivv_prop
3752 { %all darkgray
3753     linkcolor      = [rgb]{0.4 ,0.4 ,0.4 },
3754     filecolor      = [rgb]{0.4 ,0.4 ,0.4 },
3755     urlcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3756     menucolor      = [rgb]{0.4 ,0.4 ,0.4 },
3757     runcolor       = [rgb]{0.4 ,0.4 ,0.4 },
3758     %----- %-----
3759     linkbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3760     filebordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3761     urlbordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3762     menubordercolor = [rgb]{0.4 ,0.4 ,0.4 },
3763     runbordercolor = [rgb]{0.4 ,0.4 ,0.4 }
3764 }
3765
3766 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsA_prop
3767 { %dvipsnam.def
3768     linkcolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3769     filecolor      = [rgb]{1, 0, 0}, %Red
3770     urlcolor       = [rgb]{0.06, 0.46, 1}, %NavyBlue
3771     menucolor      = [rgb]{1, 0, 0}, %Red
3772     runcolor       = [rgb]{1, 0, 0}, %Red
3773     %----- %-----
3774     linkbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3775     filebordercolor = [rgb]{1, 0, 0}, %Red
3776     urlbordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3777     menubordercolor = [rgb]{1, 0, 0}, %Red
3778     runbordercolor = [rgb]{1, 0, 0} %Red
3779 }
3780
3781 \prop_const_from_keyval:Nn \c__hyp_colorscheme_szabolcsB_prop
3782 { %dvipsnam.def
3783     linkcolor      = [rgb]{0.72, 0, 0}, %BrickRed
3784     filecolor      = [rgb]{0, 1, 0}, %Green
3785     urlcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3786     menucolor      = [rgb]{0.06, 0.46, 1}, %NavyBlue
3787     runcolor       = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3788     %----- %-----
3789     linkbordercolor = [rgb]{0.72, 0, 0}, %BrickRed
3790     filebordercolor = [rgb]{0, 1, 0}, %Green
3791     urlbordercolor = [rgb]{0.64, 0.08, 0.98}, %Mulberry
3792     menubordercolor = [rgb]{0.06, 0.46, 1}, %NavyBlue
3793     runbordercolor = [rgb]{0.64, 0.08, 0.98} %Mulberry
3794 }
3795
3796
3797 \prop_const_from_keyval:Nn \c__hyp_colorscheme_phelype_prop
3798 {
3799     linkcolor      = [rgb]{0.50196, 0, 0.02353},

```



```

3800   filecolor      = [rgb]{0.07451, 0.09412, 0.46667},
3801   urlcolor      = [rgb]{0.54118, 0, 0.52941},
3802   menucolor     = [rgb]{0.44706, 0.45882, 0},
3803   runcolor      = [rgb]{0.07451, 0.46667, 0.46275},
3804   %-----      %-----
3805   linkbordercolor = [rgb]{0.701176, 0.4, 0.414118},
3806   filebordercolor = [rgb]{0.444706, 0.456472, 0.680002},
3807   urlbordercolor  = [rgb]{0.724708, 0.4, 0.717646},
3808   menubordercolor = [rgb]{0.668236, 0.675292, 0.4},
3809   runbordercolor  = [rgb]{0.444706, 0.680002, 0.67765}
3810 }
3811
3812 \prop_const_from_keyval:Nn \c__hyp_colorscheme_henryford_prop
3813 {
3814   linkcolor      = [rgb]{0,0,0},
3815   filecolor      = [rgb]{0,0,0},
3816   urlcolor       = [rgb]{0,0,0},
3817   menucolor      = [rgb]{0,0,0},
3818   runcolor       = [rgb]{0,0,0},
3819   %-----      %-----
3820   linkbordercolor = [rgb]{0,0,0},
3821   filebordercolor = [rgb]{0,0,0},
3822   urlbordercolor  = [rgb]{0,0,0},
3823   menubordercolor = [rgb]{0,0,0},
3824   runbordercolor  = [rgb]{0,0,0}
3825 }
3826 </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		
<code>\#</code>	272, 804	
<code>\\$</code>	271	
<code>\%</code>	805	
<code>\-</code>	2222	
<code>\.</code>	566, 570, 572	
@curropt commands:		box commands:
<code>\@curropt:</code>	2741	<code>\box_dp:N</code>
<code>\[</code>	2222	<code>\box_ht:N</code>
<code>\]</code>	23, 24, 25, 32, 33, 37, 47, 67,	<code>\box_new:N</code>
	74, 81, 88, 95, 110, 115, 116, 124,	<code>\box_set_dp:Nn</code>
	125, 134, 135, 136, 137, 138, 145,	<code>\box_set_ht:Nn</code>
	153, 273, 274, 955, 2743, 2808, 3145	<code>\box_use:N</code>
<code>_</code>	564, 566, 570, 572	<code>\box_use_drop:N</code>
<code>\]</code>	2222	<code>\box_wd:N</code>
		<code>\l_tmpa_box</code>
		2887, 2889, 2890,
		2891, 2893, 2926, 2940, 2941, 2942,
		2951, 2973, 2975, 2976, 2977, 2980
A		C
<code>\A</code>	564, 2222	<code>\catcode</code>
<code>\Acrobatmenu</code>	19, 166	<code>\char</code>
<code>\addcontentsline</code>	14	<code>\chardef</code>
<code>\AddToDocumentProperties</code>	381, 2217, 2320	<code>\cite</code>
<code>\AddToHook</code>	419, 444, 450	35
<code>\AddToHookNext</code>	191	clist commands:
<code>\advance</code>	2678, 2748, 2792, 2793	<code>\clist_item:nn</code>
<code>allcolors</code> (hypersetup key)	<u>1083</u>	<code>\clist_map_function:nN</code>
<code>\author</code>	2	<code>\clist_map_inline:nn</code>
		<code>\clist_map_variable:nNn</code> ..
		2740, 2806
		color commands:
		<code>\color_export:nnN</code> 42, 427, 1003, 1125
		<code>\color_select:n</code>
	 1011, 1047, 1328, 2932, 2936
		<code>\color_select:nn</code>
		1017
		<code>\color_set:nn</code>
		2, 386, 1028
		<code>\color_set:nnn</code>
		2, 385, 1034
		color names:
		<code>hyp/annot/file</code>
		547
		<code>hyp/annot/link</code>
		547
		<code>hyp/annot/menu</code>
		547
		<code>hyp/annot/run</code>
		547
		<code>hyp/annot/url</code>
		547
		<code>colorfile</code> (hypersetup key)
		<u>1083</u>
		<code>colorlink</code> (hypersetup key)
		<u>1083</u>
		<code>colorlinks</code> (hypersetup key)
		<u>1060</u>
		<code>colormenu</code> (hypersetup key)
		<u>1083</u>
		<code>colorrun</code> (hypersetup key)
		<u>1083</u>
		<code>colorscheme</code> (hypersetup key)
		1, <u>1495</u>
		<code>colorurl</code> (hypersetup key)
		<u>1083</u>
		cs commands:
		<code>\cs_generate_variant:Nn</code>
	 148, 149, 150, 383, 463,
		602, 609, 832, 1003, 1020, 1037, 2223
		<code>\cs_gset:Npn</code>
		1299
B		
<code>\b</code>	568	
<code>\begingroup</code>	202,	
	270, 368, 2584, 2708, 2737, 2749,	
	2783, 3167, 3595, 3607, 3636, 3666	
<code>\belowpdfbookmark</code>	3557	
<code>\bgroup</code>	267, 368	
<code>\BOOKMARK</code>	3560	
<code>\bookmarksetup</code>	436, 438, 439	
<code>bookmarkstype</code> (hypersetup key)	14	
bool commands:		
<code>\bool_if:NTF</code>	260,	
	289, 348, 724, 748, 768, 788, 799,	
	835, 901, 966, 1044, 1054, 1307, 1318	
<code>\bool_if:nTF</code> 916, 1992, 2017, 3302, 3409		
<code>\bool_lazy_and:nnTF</code>	416	
<code>\bool_lazy_or:nnTF</code>	1337, 1386	
<code>\bool_new:N</code>	196, 197, 550, 554, 558	
<code>\bool_set_true:N</code>	559	
<code>bordercolormodel</code> (hypersetup key) <i>14</i> , <u>1104</u>		

<code>\cs_gset_eq:NN</code>	402, 413	<code>destlabel</code> (hypersetup key)	14
<code>\cs_if_exist_p:N</code>	417	dim commands:	
<code>\cs_new:Npn</code>	110, 633	<code>\dim_to_decimal_in_bp:n</code>	12, 154
<code>\cs_new_protected:Npn</code>	202,	<code>\do</code>	2741
	298, 315, 332, 358, 378, 393, 404,	<code>\documentclass</code>	25
	457, 578, 582, 586, 590, 603, 605,	<code>\DocumentMetadata</code>	
	610, 674, 682, 691, 695, 717, 722, 1, 3, 11, 15, 16, 19, 24, 33, 40	
	746, 766, 786, 833, 899, 964, 1004,	<code>draft</code> (hypersetup key)	1519
	1015, 1021, 1032, 1243, 2224, 2306		
<code>\cs_set:Nn</code>	208, 250	E	
<code>\cs_set_eq:NN</code>	154,	<code>\edef</code> 2586, 2751, 2787, 2794, 3130, 3601, 3672	
	159, 160, 424, 641, 643, 804, 805, 1553	<code>\EdefEscapeName</code>	2805, 2809
<code>\cs_set_protected:Npn</code>		<code>\egroup</code>	371
.....	152, 173, 620, 955	<code>\else</code>	2583, 2704, 2758,
<code>\csname</code>	2755, 3137,	2762, 2765, 2782, 2790, 3118, 3123,	
3148, 3627, 3637, 3680, 3681, 3682,		3140, 3146, 3158, 3162, 3179, 3183,	
3683, 3684, 3685, 3686, 3687, 3688		3187, 3191, 3195, 3199, 3203, 3207,	
<code>\currentpdfbookmark</code>	3553	3211, 3215, 3231, 3235, 3335, 3336,	
		3337, 3339, 3342, 3346, 3350, 3356,	
		3373, 3377, 3381, 3387, 3391, 3458,	
		3459, 3461, 3464, 3468, 3493, 3494,	
		3496, 3499, 3503, 3570, 3639, 3660	
		<code>\endcsname</code>	2769, 3114, 3137,
		3148, 3627, 3637, 3680, 3681, 3682,	
		3683, 3684, 3685, 3686, 3687, 3688	
		<code>\endgroup</code>	276, 279,
		369, 371, 2587, 2728, 2751, 2797,	
		2800, 3174, 3601, 3613, 3636, 3672	
		<code>\endinput</code>	3571
		exp commands:	
		<code>\exp_args:Ne</code>	842, 2334, 2578, 3222
		<code>\exp_args:Nne</code>	2349, 2531
		<code>\exp_args:Nno</code>	317, 334
		<code>\exp_args:NNO</code>	300, 360, 2292
		<code>\exp_args:NNV</code>	1977, 2008, 2253
		<code>\exp_args:NnV</code>	676, 685
		<code>\exp_args:No</code>	
	 709, 713, 2870, 2913, 2961, 2996	
		<code>\exp_args:NV</code>	657, 664
		<code>\exp_not:N</code>	955
		<code>\exp_not:n</code>	
		1434, 1468, 1560, 1615, 1636, 1673,	
		1693, 1716, 1760, 1793, 1818, 1869,	
		1894, 1923, 1972, 2049, 2077, 2104,	
		2124, 2370, 2476, 2504, 2514, 2525	
		<code>\expandafter</code>	273, 2709, 2743,
		2752, 2754, 2755, 2808, 3131, 3147,	
		3571, 3597, 3598, 3636, 3637, 3659,	
		3661, 3668, 3669, 3680, 3681, 3682,	
		3683, 3684, 3685, 3686, 3687, 3688	
		<code>\ExplSyntaxOff</code>	
		15, 2615, 2668, 3112, 3325, 3447, 3546	
		<code>\ExplSyntaxOn</code>	
		. 13, 16, 2616, 2673, 3219, 3360, 3483	
D			
<code>\d</code>	566, 570, 572		
<code>debug</code> (hypersetup key)	1519		
<code>\DeclareRobustCommand</code>			
... 255, 283, 306, 323, 342, 371, 3579			
<code>\def</code>	12, 158, 368, 369, 371,		
615, 2547, 2551, 2558, 2576, 2605,			
2618, 2676, 2683, 2696, 2697, 2700,			
2701, 2714, 2715, 2731, 2732, 2735,			
2736, 2774, 2803, 2853, 2867, 2868,			
2880, 2908, 2910, 2911, 2921, 2956,			
2958, 2959, 2970, 2989, 2991, 2992,			
2994, 2995, 3004, 3048, 3115, 3125,			
3127, 3132, 3144, 3155, 3166, 3177,			
3220, 3228, 3240, 3292, 3326, 3361,			
3399, 3449, 3484, 3521, 3528, 3535,			
3542, 3574, 3592, 3606, 3618, 3619,			
3642, 3649, 3657, 3680, 3681, 3682,			
3683, 3684, 3685, 3686, 3687, 3688			
<code>\DefaultHeightofCheckBox</code>	2995		
<code>\DefaultHeightofChoiceMenu</code>	2736		
<code>\DefaultHeightofReset</code>	2959		
<code>\DefaultHeightofSubmit</code>	2911		
<code>\DefaultHeightofText</code>	2705		
<code>\DefaultHeightofTextMultiline</code>	2703		
<code>\DefaultOptionsofCheckBox</code>	2998		
<code>\DefaultOptionsofPushButton</code>	2872		
<code>\DefaultOptionsofReset</code>	2963		
<code>\DefaultOptionsofSubmit</code>	2915		
<code>\DefaultOptionsofText</code>	2710		
<code>\DefaultWidthofCheckBox</code>	2994		
<code>\DefaultWidthofChoiceMenu</code>	2735		
<code>\DefaultWidthofReset</code>	2958		
<code>\DefaultWidthofSubmit</code>	2910		
<code>\DefaultWidthofText</code>	2700		

extension (hypersetup key)	14, 1537	hidelink (hypersetup key)	1471
F			
<code>\fbox</code>	2844, 2932, 2936	hidelinks (hypersetup key)	1471
<code>\fi</code>	12, 2572, 2599, 2602, 2611, 2706, 2714, 2715, 2746, 2764, 2767, 2768, 2774, 2777, 2786, 2788, 2795, 2798, 3120, 3142, 3152, 3154, 3160, 3164, 3171, 3172, 3181, 3185, 3189, 3193, 3197, 3201, 3205, 3209, 3213, 3217, 3233, 3237, 3335, 3336, 3337, 3344, 3348, 3352, 3354, 3356, 3375, 3379, 3383, 3387, 3394, 3458, 3459, 3466, 3470, 3472, 3493, 3494, 3501, 3505, 3507, 3572, 3577, 3629, 3630, 3641, 3645, 3662	hidemenu (hypersetup key)	1471
file (hypersetup key)	10, 1568	hiderun (hypersetup key)	1471
file commands:		hideurl (hypersetup key)	1471
<code>\file_input:n</code>	14	hook commands:	
fileborderstyle (hypersetup key)	14, 1197	<code>\hook_gput_code:nnn</code> 395, 406, 1040, 1050, 1303, 1314
filecolor (hypersetup key)	1083	<code>\hook_new:n</code>	384, 577, 673
final (hypersetup key)	1519	<code>\hook_new_pair:nn</code> 253, 281, 305, 322, 340
<code>\fontencoding</code>	2821, 3018, 3030	<code>\hook_use:n</code> 257, 277, 286, 302, 309, 319, 326, 336, 345, 364, 594, 679, 688, 733, 757
<code>\fontfamily</code>	2820, 3017, 3029	<code>\href</code>	4, 5, 20, 255
<code>\fontseries</code>	2822, 3019, 3031	<code>\hreflaunch</code>	5
<code>\fontshape</code>	2823, 3020, 3032	<code>\hrefpdf</code>	5, 20, 306, 376
fp commands:		<code>\hrefrun</code>	5, 323, 377
<code>\fp_eval:n</code>	2147, 2175, 2191	<code>\hrefurl</code>	5, 20, 283, 375
G			
<code>\gdef</code>	273, 274	<code>\hss</code>	3586
<code>\GetDocumentProperties</code>	3	hyp commands:	
<code>\global</code>	2678, 3110	<code>\l_hyp_annot_colorfile_bool</code>	8, 548
group commands:		<code>\l_hyp_annot_colorlink_bool</code>	8, 548
<code>\group_begin:</code>	258, 287, 310, 327, 346, 592, 730, 740, 754, 761, 793, 821, 840, 889, 906, 948, 971, 997, 1046, 1310, 1327, 2455, 2819, 2869, 2912, 2960, 2993, 3016, 3028	<code>\l_hyp_annot_colormenu_bool</code>	8, 548
<code>\group_end:</code>	301, 318, 335, 363, 599, 736, 740, 773, 777, 818, 821, 886, 889, 946, 948, 995, 997, 1056, 1330, 1333, 2462, 2826, 2905, 2953, 2986, 3023, 3035, 3045	<code>\l_hyp_annot_colorurl_bool</code>	8, 548
H			
<code>\hbox</code>	368, 3581	<code>\l_hyp_annot_ocgcolorfile_bool</code>	8, 552
hbox commands:		<code>\l_hyp_annot_ocgcolorlink_bool</code>	8, 552
<code>\hbox_overlap_right:n</code>	1324	<code>\l_hyp_annot_ocgcolormenu_bool</code>	8, 552
<code>\hbox_set:Nn</code>	2887, 2926, 2973	<code>\l_hyp_annot_ocgcolorurl_bool</code>	8, 552
<code>\hbox_set:Nw</code>	1311	<code>\l_hyp_annot_ocgcolorrun_bool</code>	8, 552
<code>\hbox_set_end:</code>	1320	<code>\l_hyp_annot_ocgcolormenu_bool</code>	8, 552
<code>\hbox_set_to_wd:Nnn</code>	2173	<code>\l_hyp_annot_ocgcolorurl_bool</code>	8, 552
hidefile (hypersetup key)	1471	<code>\l_hyp_current_dest_name_tl</code> 672, 731, 755
		hyp internal commands:	
		<code>\g__hyp_AcroForm_CoFields_prop</code>	2542, 2560, 2629, 2631, 2646
		<code>\g__hyp_AcroForm_Fields_prop</code>	2543, 2553, 2624
		<code>\l__hyp_annot_GoTo_bool</code>	724, 748, 768
		<code>\l__hyp_annot_GoTo_bool_uuuu\l__</code> <code>hyp_annot_URI_bool_uuuu\l__</code> <code>hyp_annot_GoToR_bool_uuuu\l__</code> <code>hyp_annot_Named_bool_uuuu\l__</code> <code>hyp_annot_Launch_bool</code>	556
		<code>\l__hyp_annot_GoToR_bool</code>	835
		<code>\l__hyp_annot_Launch_bool</code>	901
		<code>\l__hyp_annot_Named_bool</code>	966
		<code>\c__hyp_annot_types_seq</code> 477, 548, 552, 1083, 1384, 1482
		<code>\l__hyp_annot_URI_bool</code>	788
		<code>\g__hyp_bordercolormodel_str</code> 429, 547 , 1108, 1127

__hyp_check_link_nesting:TF ...	\l__hyp_href_run_parameter_tl ..
..... 633, 641, 201, 222, 334
643, 727, 751, 770, 791, 838, 904, 969	__hyp_href_url_aux:n 355, 358
__hyp_citebordercolor_hook_-	__hyp_href_url_aux:nn 296, 298
init: 391, 404, 413	\l__hyp_href_url_encode_bool ...
__hyp_citecolor_hook_init: 196, 207, 245, 260, 289, 348
..... 389, 393, 402	__hyp_href_url_format:
__hyp_clist_display:n . 110, 117, 126 202, 208, 250, 361
__hyp_color_select:n	\l__hyp_href_url_ismap_bool
..... 42, 1004, 1004, 1020 197, 221, 799
__hyp_color_select_aux:wn	\l__hyp_href_url_protocol_tl ...
..... 1004, 1008, 1015 198, 209, 248, 300, 362
__hyp_color_set:nn 42, 389,	__hyp_if_outer_link:
391, 426, 1021, 1021, 1037, 1088, 1124	__hyp_if_outer_link:TF
__hyp_color_set_aux:nwn	643
..... 1021, 1025, 1032	\l__hyp_link_Contents_tl
\c__hyp_colorscheme_daleif_prop 3721 512, 521, 523, 530, 532, 539, 541
\c__hyp_colorscheme_henryford_-	__hyp_link_goto_begin:nw
prop 3812 695, 734, 758
\c__hyp_colorscheme_julian_prop 3736	__hyp_link_goto_end: .. 717, 735, 772
\c__hyp_colorscheme_phelype_prop	\g__hyp_linknestlevel_int
..... 3797 626, 629, 726, 742, 750, 779,
\c__hyp_colorscheme_szabolcsA_-	790, 823, 837, 891, 903, 949, 968, 998
prop 3766	g__hyp_linknestlevel_int
\c__hyp_colorscheme_szabolcsB_-	626
prop 3781	\c__hyp_map_annot_hyp_prop 477
\c__hyp_colorscheme_tivv_prop . 3751	\c__hyp_map_hyp_annot_prop
\l__hyp_dest_box 477, 1038,
33,	1111, 1151, 1178, 1186, 1197, 1224,
561, 659, 660, 661, 2173, 2189, 2196	1232, 1301, 1409, 1445, 1455, 1568
\l__hyp_dest_name_tmpa_tl .. 470,	__hyp_ocg_init: 1243, 1243, 1299, 1309
698, 699, 705, 709, 711, 714, 858, 872	\l__hyp_optlang_regex 2221, 2222, 2254
\l__hyp_dest_pdfremotestartview_-	l__hyp_page/Trans
tl 496, 867, 1980, 1984	575
\g__hyp_dest_pdfstartpage_tl ...	__hyp_PageLabels_gpush: 610, 610, 623
... 496, 1991, 1993, 2000, 2018, 2025	\l__hyp_para_tmpa_tl 473, 915, 918, 929
\g__hyp_dest_pdfstartview_tl ...	\l__hyp_para_tmpa_tl_uuu\l__hyp_-
496, 1993, 2000, 2011, 2015, 2018, 2025	text_tmpa_str_uuu\g__hyp_text_-
\l__hyp_dest_pdfview_tl	tmpa_str
..... 546, 677, 686,	470
2142, 2145, 2152, 2155, 2156, 2157,	__hyp_property_record:nn
2158, 2159, 2160, 2165, 2169, 2205 456, 457, 463, 2679
\c__hyp_dest_startview_regex ...	__hyp_secondoftwoewithopt:wnn ..
..... 562, 1978, 2009 374, 375, 376, 377
\c__hyp_dest_undefined_tl	__hyp_setup_info_date_key:nn ..
..... 476, 704, 705 2306, 2339, 2340
__hyp_destination:nn	__hyp_setup_info_key:nn .. 2224,
..... 33, 646, 646, 677, 686	2299, 2300, 2301, 2302, 2304, 2305
\l__hyp_filename_tmpa_tl ... 470,	__hyp_store_metadata:nn
841, 843, 848, 849, 854, 910, 911, 929 378, 383, 1587, 2231,
__hyp_href_pdf_aux:nn 312, 315	2274, 2319, 2334, 2343, 2359, 2429
\l__hyp_href_pdf_destination_tl	__hyp_text_cleanup:N .. 582, 582, 596
..... 199, 210, 249, 317	\l__hyp_text_enc_dest_print_tl .
\l__hyp_href_pdf_page_tl 200, 218, 866 499, 857
__hyp_href_run_aux:nn 329, 332	\l__hyp_text_enc_dest_tl
 33, 499, 653, 710
	\l__hyp_text_enc_file_print_tl .
 499, 909

\l__hyp_text_enc_info_print_tl	499, 520, 529, 538, 607	colorscheme	1, 1495
\l__hyp_text_enc_para_print_tl	499, 914	colorurl	1083
\l__hyp_text_enc_uri_print_tl	262, 265, 291, 294, 350, 353, 499, 796, 1579	debug	1519
__hyp_text_pdfstring:nnN	518, 527, 536, 590, 590, 602, 604, 607, 651, 708, 794, 855, 907, 912, 1579	destlabel	14
__hyp_text_pdfstring_info:nN	605, 605, 609, 2257, 2260, 2280, 2289	draft	1519
__hyp_text_purify:nN	578, 578, 595	extension	14, 1537
__hyp_text_string_from_unicode:nN	586, 586, 597	file	10, 1568
\g__hyp_text_tmpa_str	475, 598, 600	fileborderstyle	14, 1197
\l__hyp_text_tmpa_str	474, 595, 596, 597, 598	filecolor	1083
\l__hyp_tmpa_box	464, 1311, 1324, 1329	final	1519
\l__hyp_tmpa_int	464	hidefile	1471
\l__hyp_tmpa_seq	464, 1978, 1980, 2009, 2011, 2132, 2133, 2137, 2139, 2163, 2170, 2171, 2172, 2181, 2183, 2193, 2198, 2254, 2255, 2260, 2633, 2635, 2641	hidelink	1471
\l__hyp_tmpa_str	464, 2257, 2260, 2262, 2264, 2280, 2281, 2283, 2289, 2290, 2294	hidelinks	1471
\l__hyp_tmpa_tl	464, 654, 658, 666, 1128, 1132, 1579, 1580, 1585, 1976, 1978, 2007, 2009, 2139, 2140, 2147, 2230, 2231, 2232, 2252, 2257, 2273, 2274, 2280	hidemenu	1471
\l__hyp_tmpb_tl	464, 2252, 2254	hiderun	1471
\l__hyp_uri_tmpa_tl	470, 797, 798	hideurl	1471
hyp/anchor	672	hypertextnames	1537
hyp/annot/file (color name)	547	link	10, 1568
hyp/annot/link (color name)	547	linkborder	14
hyp/annot/menu (color name)	547	linkborderstyle	14, 1197
hyp/annot/run (color name)	547	linkcolor	1083
hyp/annot/url (color name)	547	linkfileprefix	1537
hyp/text/pdfstring	577	linktoc	1537
\hypercalcbp	12, 19, 154	linktocpage	1537
\HyperDestNameFilter	13, 652, 709	localanchorname	1537
\hypersetup	2, 3, 6, 9, 10, 14, 19, 70, 102, 173	menu	10, 1568
\hypersetup keys:		menuborder	14
allcolors	1083	menuborderstyle	14, 1197
bookmarkstype	14	menucolor	1083
bordercolormodel	14, 1104	naturalnames	1537
colorfile	1083	nested-links	10
colorlink	1083	nesting	14
colorlinks	1060	ocgcolorfile	1337
colormenu	1083	ocgcolorlink	1337
colorrun	1083	ocgcolorlinks	1337
		ocgcolormenu	1337
		ocgcolorrun	1337
		ocgcolorurl	1337
		pageanchor	1537
		pdfauthor	2221
		pdfborder	14
		pdfborderstyle	14, 1197
		pdfcreationdate	15, 2306
		pdfcreator	2221
		pdfencoding	1509
		pdfinfo	2373
		pdfkeywords	2221
		pdflang	15, 2210
		pdflinkmargin	15
		pdfmetadate	15, 2306
		pdfmoddate	15, 2306
		pdfproducer	2221
		pdfremotestartview	12
		pdfstartview	12

pdfsubject	2221	2210, 2226, 2269, 2308, 2323, 2341,
pdftitle	2221	2345, 2373, 2427, 2433, 2446, 2466
pdftrapped	2345	\l_keys_key_str 184, 2294, 2531
pdfversion	1509	\keys_set:nn 176,
pdfview	12, 2128	213, 232, 259, 288, 311, 328, 347,
plainpages	1537	397, 408, 1501, 1508, 2350, 2377,
run	10, 1568	2380, 2381, 2382, 2383, 2456, 2531
runborder	14	\keys_set_known:nn 2534
runborderstyle	14, 1197	\kvsetkeys 175, 2620
runcolor	1083	
unicode	1509	
url	10, 1568	
urlborder	14	
urlborderstyle	14, 1197	
urlcolor	1083	
verbose	1519	
hypertextnames (hypersetup key)	1537	
I		
\ifcase	3335, 3458, 3493	
\ifcsname	3114	
\ifdim	2746, 2785, 2786	
\ifnum	3168, 3335, 3341, 3372, 3458,	
	3463, 3493, 3498, 3625, 3643, 3658	
\iftrue	3627	
\ifx	2582, 2715, 2775,	
	3145, 3157, 3161, 3178, 3182, 3186,	
	3190, 3194, 3198, 3202, 3206, 3210,	
	3214, 3230, 3234, 3336, 3337, 3345,	
	3349, 3356, 3376, 3380, 3387, 3388,	
	3459, 3467, 3494, 3502, 3575, 3637	
\immediate	2566, 2569	
int commands:		
\int_compare:nNnTF	629, 2137,	
	2163, 2845, 3259, 3311, 3420, 3442	
\int_compare_p:nNn	3304, 3411	
\int_eval:n	865	
\int_gdecr:N	742, 779, 823, 891, 949, 998	
\int_gincr:N	726, 750, 790, 837, 903, 968	
\int_max:nn	866	
\int_new:N	468, 626	
iow commands:		
\iow_newline:	3063, 3069, 3082	
K		
\kern	3582	
keys commands:		
\keys_define:nn	179, 187, 205, 243, 387,	
	433, 637, 1060, 1085, 1092, 1104,	
	1113, 1138, 1153, 1172, 1199, 1218,	
	1341, 1355, 1367, 1390, 1396, 1411,	
	1440, 1471, 1484, 1495, 1509, 1519,	
	1524, 1537, 1549, 1570, 1575, 2128,	
	2210, 2226, 2269, 2308, 2323, 2341,	
	2345, 2373, 2427, 2433, 2446, 2466	
	\l_keys_key_str 184, 2294, 2531	
	\keys_set:nn 176,	
	213, 232, 259, 288, 311, 328, 347,	
	397, 408, 1501, 1508, 2350, 2377,	
	2380, 2381, 2382, 2383, 2456, 2531	
	\keys_set_known:nn 2534	
	\kvsetkeys 175, 2620	
L		
\label	9, 14	
\LayoutCheckField	3006	
\LayoutChoiceField	2778	
\LayoutPushButtonField	2882, 2899	
\LayoutTextField	2716	
\leavevmode	2717, 2812, 3580	
legacy commands:		
\legacy_if:nTF	161, 442, 452,	
	618, 2658, 2878, 2919, 2969, 3002, 3282	
\let	375,	
	376, 377, 744, 813, 825, 882, 893,	
	942, 951, 1000, 2545, 2546, 2573,	
	2669, 2670, 2671, 2698, 2699, 2733,	
	2734, 2776, 3110, 3117, 3119, 3124,	
	3141, 3175, 3591, 3605, 3608, 3617,	
	3628, 3635, 3638, 3640, 3648, 3656	
link (hypersetup key)	10, 1568	
linkborder (hypersetup key)	14	
linkborderstyle (hypersetup key)	14, 1197	
linkcolor (hypersetup key)	1083	
linkfileprefix (hypersetup key)	1537	
linktoc (hypersetup key)	1537	
linktocpage (hypersetup key)	1537	
localanchorname (hypersetup key)	1537	
\long	12	
M		
\MakeButtonField	2887, 2902, 2926, 2973, 2984	
\MakeChoiceField	2862	
\MakeFieldObject	2536	
\MakeRadioField	2844	
\MakeTextField	2725	
\mbox	1321	
menu (hypersetup key)	10, 1568	
menuborder (hypersetup key)	14	
menuborderstyle (hypersetup key)	14, 1197	
menucolor (hypersetup key)	1083	
mode commands:		
\mode_if_horizontal:TF	648, 670	
\mode_leave_vertical:	256, 285,	
	308, 325, 344, 697, 807, 874, 931,	
	974, 2856, 2884, 2901, 2923, 2965, 3008	

msg commands:

- `\msg_error:nn` 418, 2898, 2983
- `\msg_info:nnn` 2238, 2243
- `\msg_line_context:` 75
- `\g_msg_module_name_prop` 17
- `\msg_new:nnn` . 43, 50, 55, 59, 63, 70, 77, 84, 91, 98, 104, 111, 120, 130, 141
- `\msg_new:nnnn` 18, 28
- `\msg_warning:nn` 163, 1515
- `\msg_warning:nnn` 172, 701, 992
- `\msg_warning:nnnn` 1359, 1400, 1613, 1657, 1691, 1714, 1776, 1805, 1839, 1854, 1867, 1880, 1905, 1940, 1956, 1983, 2014, 2047, 2061, 2088, 2122, 2204
- `\msg_warning:nnnnn` 183, 1431, 1465, 1557, 1633, 1670, 1757, 1790, 1815, 1891, 1918, 1969, 2074, 2101, 2367, 2473, 2492, 2511, 2522

N

- `naturalnames` (hypersetup key) [1537](#)
- `nested-links` (hypersetup key) [10](#)
- `nesting` (hypersetup key) [14](#)
- `\newcommand` 157, 3554, 3556, 3558
- `\newcount` 2674, 2852
- `\NewDocumentCommand` 2536
- `\NewExpandableDocumentCommand` 374
- `\newlength` 3573
- `\noexpand` 2752, 2753, 2754, 2755, 3128
- `\nolinkurl` 4

O

- `ocgcolorfile` (hypersetup key) [1337](#)
- `ocgcolorlink` (hypersetup key) [1337](#)
- `ocgcolorlinks` (hypersetup key) [1337](#)
- `ocgcolormenu` (hypersetup key) [1337](#)
- `ocgcolorrun` (hypersetup key) [1337](#)
- `ocgcolorurl` (hypersetup key) [1337](#)

P

- `pageanchor` (hypersetup key) [1537](#)
- `\paperwidth` 4
- `\PassOptionsToPackage` 454, 1529, 1534

pdf commands:

- `\pdf_bdcobject:nn` 1323, 1326
- `\pdf_destination:nn` 28, 148, 665
- `\pdf_destination:nnnn` 658
- `\pdf_emc:` 1325, 1331
- `\pdf_name_from_unicode_e:n` 238, 845, 976
- `\pdf_object_if_exist:nTF` 843
- `\pdf_object_new:n` 1245, 1246, 1247, 1248, 3050, 3051, 3052
- `\pdf_object_ref:n` 149, 854, 1251, 1253, 1275, 1276, 1279, 1283, 1288, 1293, 1298, 2651, 2653, 3108
- `\pdf_object_ref_last:` 876, 2461
- `\pdf_object_unnamed_write:nn` 875, 2457
- `\pdf_object_write:nnn` 1249, 1255, 1265, 1277, 3053, 3095, 3102
- `\pdf_pageobject_ref:n` 150, 2000, 2025, 2687
- `\pdf_string_from_unicode:nN` 588
- `\pdf_version:` 1661, 1780, 1809, 1843, 1858, 1884, 1909, 1944, 1960, 2065, 2092
- `\pdf_version_compare:NnTF` 1651, 1742, 1764, 1800, 1833, 1848, 1874, 1899, 1927, 1950, 2055, 2082
- `\pdf_version_compare_p:Nn` 920, 1338, 1387
- `\pdf_version_major:` 160, 1339, 1362, 1388, 1404
- `\pdf_version_minor:` 159, 1362, 1404

pdfannot commands:

- `\pdfannot_box:nnnn` 2720, 2829, 2859, 2888, 2939, 2974, 3038
- `\pdfannot_box_ref_last:` 2580, 2596
- `\pdfannot_dict_put:nnn` 522, 531, 540, 808, 876, 932, 977, 1129, 1164, 1188, 1210, 1234, 1416, 1447
- `\pdfannot_dict_remove:nn` 1119, 1159, 1180, 1205, 1226, 1424, 1457
- `\pdfannot_link:nnn` 809, 877, 933, 978
- `\pdfannot_link_goto_begin:nw` 714
- `\pdfannot_link_goto_end:` 719
- `\pdfannot_link_margin:n` 8, 1695
- `\c_pdfannot_link_types_seq` 556

pdfauthor (hypersetup key) [2221](#)

\pdfbookmark 3559

pdfborder (hypersetup key) 14

pdfborderstyle (hypersetup key) [14](#), [1197](#)

pdfcreationdate (hypersetup key) [15](#), [2306](#)

pdfcreator (hypersetup key) [2221](#)

\pdfdest 12

pdfdict commands:

- `\pdfdict_new:n` 575, 782, 828, 895, 960
- `\pdfdict_put:nnn` 230, 237, 576, 783, 784, 798, 801, 829, 830, 845, 851, 862, 872, 896, 897, 911, 926, 961, 962, 975, 1721, 1722, 1726, 1727, 2470, 2480, 2484, 2487, 2489, 2508, 2517, 2519
- `\pdfdict_remove:nn` 227, 923, 1731, 1732
- `\pdfdict_use:n` 808, 875, 932, 938, 977, 982, 2459

S	
<code>\selectfont</code>	2824, 3021, 3033
seq commands:	
<code>\seq_const_from_clist:Nn</code>	477
<code>\seq_count:N</code>	2137, 2163
<code>\seq_get_right:NN</code>	2139
<code>\seq_if_empty:NTF</code>	2255
<code>\seq_item:Nn</code>	1980, 2011, 2133, 2181, 2183, 2193, 2198, 2260
<code>\seq_map_inline:Nn</code>	548, 552, 556, 1083, 1384, 1482, 2641
<code>\seq_new:N</code>	467
<code>\seq_put_right:Nn</code>	2170, 2171, 2172, 2633
<code>\seq_set_split:Nnn</code>	2132
<code>\seq_sort:Nn</code>	2635
<code>\setbox</code>	368
<code>\SetKeys</code>	34
<code>\setlength</code>	3594, 3665
<code>\setpdfinkmargin</code>	8
<code>\settowidth</code>	2745
socket commands:	
<code>\socket_assign_plug:nn</code> .	543, 544, 545
<code>\socket_new:nn</code>	513, 514, 515
<code>\socket_new_plug:nmn</code> ...	516, 525, 534
<code>\socket_use:nn</code>	803, 859
<code>\socket_use:nmn</code>	732, 756
sort commands:	
<code>\sort_return_same:</code>	2639
<code>\sort_return_swapped:</code>	2638
<code>\space</code>	2811, 3338, 3355, 3356, 3386, 3387, 3460, 3492, 3495
<code>\spacefactor</code>	648, 670
<code>\stockwidth</code>	4
str commands:	
<code>\c_colon_str</code>	955
<code>\c_hash_str</code>	804
<code>\c_percent_str</code>	805
<code>\str_case:nnTF</code>	2234, 3517
<code>\str_case_e:nnTF</code>	2133
<code>\str_compare:nNnTF</code>	2637
<code>\str_gset:Nn</code>	1108
<code>\str_gset_eq:NN</code>	598
<code>\str_head:n</code>	2356, 2361
<code>\str_if_eq:nnTF</code>	655, 2262, 2281, 2290, 3222
<code>\str_if_eq_p:nn</code>	918, 1339, 1388
<code>\str_lowercase:n</code>	2133, 2274, 2334, 2357, 2362
<code>\str_new:N</code>	469, 474, 475, 547
<code>\str_set:Nn</code>	580
<code>\str_set_convert:Nnnn</code>	9
<code>\str_set_eq:NN</code>	600
<code>\str_tail:n</code>	2357, 2362
<code>\str_uppercase:n</code> ...	2350, 2356, 2361
<code>\string</code>	2567, 2570, 2591, 2609, 3133, 3137, 3148
<code>\subpdfbookmark</code>	3555
sys commands:	
<code>\c_sys_backend_str</code>	3517
T	
TeX and L ^A T _E X 2 _ε commands:	
<code>\@Listbox</code>	2796, 2853
<code>\@Radio</code>	2781, 2803
<code>\@writetorep</code>	3550
<code>\@BOOKMARK</code>	3561
<code>\@CheckBox</code>	2989
<code>\@ChoiceMenu</code>	2731
<code>\@Form</code>	2618
<code>\@PushButton</code>	2867
<code>\@Reset</code>	2956
<code>\@Submit</code>	2908
<code>\@TextField</code>	2696
<code>\@bookmarksopenlevel</code>	1596
<code>\@bsphack</code>	459
<code>\@chapter</code>	3617, 3618
<code>\@currDisplay</code>	2745, 2811
<code>\@currValue</code>	2809, 2839
<code>\@currentHref</code>	3598, 3611, 3622, 3652, 3669
<code>\@curopt</code>	2730, 2740, 2743, 2806, 2808
<code>\@empty</code>	424, 2545, 2546, 2582, 2669, 2670, 2671, 2698, 2699, 2715, 3178, 3182, 3186, 3190, 3194, 3198, 3202, 3206, 3210, 3214, 3356, 3387, 3388
<code>\@endForm</code>	2669
<code>\@esphack</code>	461
<code>\@firstofone</code> ...	2573, 3119, 3124, 3141
<code>\@firstoftwo</code>	3659
<code>\@for</code>	2741
<code>\@gobbletwo</code>	3608
<code>\@hyper@launch</code>	40, 955
<code>\@ifnextchar</code>	267
<code>\@ifundefined</code> ..	3133, 3589, 3616, 3626
<code>\@mainaux</code>	2566, 2569, 2589, 2608
<code>\@mkboth</code>	3608
<code>\@ne</code>	2678
<code>\@nil</code>	3131, 3144, 3150
<code>\@part</code>	3635, 3642
<code>\@pdfauthor</code>	23
<code>\@pdfborder</code>	155
<code>\@pdfborderstyle</code>	156
<code>\@savsf</code>	648, 670
<code>\@schapter</code>	3605, 3606
<code>\@secondoftwo</code>	3661
<code>\@sect</code>	3656, 3657
<code>\@spart</code>	3648, 3649

\@ssect	3591, 3592	\Fld@onenter@code	3210, 3212
\@tempdima	2739, 2746, 2748, 2785, 2786, 2787, 2791, 2792, 2793, 2794	\Fld@onexit@code	3214, 3216
\@tempdimb	2745, 2746	\Fld@onfocus@code	3194, 3196
\@typeset@protect	3575	\Fld@onmousedown@code	3202, 3204
\BKM@color	424, 430	\Fld@onmouseup@code	3206, 3208
\c@secnumdepth	3625, 3643, 3658	\Fld@pageobjref	2671, 2683, 3113, 3244, 3296, 3329, 3365, 3403, 3452, 3487
\calc@bm@number	3567	\Fld@radiosymbol	3384
\check@bm@number	3566	\Fld@rotation	3259, 3261, 3304, 3311, 3313, 3335, 3341, 3343, 3372, 3374, 3411, 3420, 3422, 3458, 3463, 3465, 3493, 3498, 3500
\define@key	421	\Fld@submitflags	3479
\Fld@additionalactions	3220, 3289, 3322, 3358, 3395, 3441, 3481, 3510	\Fld@validate@code	3186, 3188
\Fld@align	3248, 3333, 3407	\Fld@value	2699, 2715, 2734, 2775, 2776, 3440
\Fld@altname	3157, 3159, 3230, 3232	\Fld@width	2700, 2714, 2721, 2725, 2735, 2774, 2785, 2787, 2830, 2844, 2860, 2862, 2880, 2910, 2921, 2958, 2970, 2994, 3004, 3039
\Fld@annotflags	3243, 3295, 3328, 3364, 3402, 3451, 3486	\Fld@X@additionalactions	3177, 3222, 3225
\Fld@annotnames	3228, 3245, 3297, 3330, 3366, 3404, 3453, 3488	\Form@action	3477
\Fld@bcolor	3267, 3269, 3349, 3351, 3380, 3382, 3415, 3428, 3430	\H@old@part	3635, 3646
\fld@bcolor	3337	\H@old@schapter	3605, 3614
\Fld@bordercolor	3263, 3265, 3306, 3315, 3317, 3336, 3345, 3347, 3376, 3378, 3413, 3424, 3426, 3459, 3467, 3469, 3494, 3502, 3504	\H@old@ssect	3656, 3667, 3676
\Fld@borderstyle	3249, 3301, 3334, 3370, 3408, 3457, 3508	\H@old@spart	3648, 3654
\Fld@borderwidth	2792, 2793, 3249, 3301, 3334, 3370, 3408, 3457, 3508	\H@old@ssect	3591, 3596
\Fld@calculate@code	2582, 3190, 3192	\href@	267, 273
\Fld@calculate@sortkey	2593	\href@split	273, 274
\Fld@cbsymbol	3271	\Hy@@escapeform	3131, 3144, 3150
\Fld@charsize	2791, 3275, 3355, 3386, 3436, 3492	\Hy@@SectionAnchor	3576, 3579
\Fld@checkequals	2743, 2808	\Hy@abspage	622
\Fld@choices	3357	\Hy@activeanchorfalse	693
\Fld@color	3276, 3278, 3356, 3387, 3437	\Hy@activeanchortrue	684
\Fld@default	2698, 2715, 2733, 2776, 2805, 2992, 3388, 3392, 3393, 3439	\Hy@AtBeginDocument	2547, 2563
\Fld@flags	3247, 3299, 3332, 3368, 3406, 3455, 3490	\Hy@bookmarkstype	1597
\Fld@format@code	3182, 3184	\Hy@chapapp	3609, 3620
\Fld@height	2701, 2722, 2725, 2736, 2794, 2831, 2844, 2860, 2862, 2911, 2959, 2995, 3040	\Hy@colorlink	36
\Fld@keystroke@code	3178, 3180	\Hy@currentbookmarklevel	3548
\Fld@listcount	2804, 2810, 2845, 2852	\Hy@DisableOption	165
\Fld@mappingname	3161, 3163, 3234, 3236	\Hy@drafttrue	1528
\Fld@maxlen	3442, 3444	\Hy@escapeform	2719, 2814, 2858, 2886, 2925, 2972, 3010, 3115, 3122, 3125
\Fld@menulength	2738, 2744, 2791	\Hy@escapestring	3117, 3119, 3124, 3127, 3128, 3131, 3132, 3141, 3144, 3180, 3184, 3188, 3192, 3196, 3200, 3204, 3208, 3212, 3216, 3271, 3321, 3384, 3439, 3440, 3477
\Fld@name	2697, 2732, 2868, 2991, 3156, 3229	\Hy@finaltrue	1533
\Fld@onblur@code	3198, 3200	\Hy@FormObjects	2623, 3048, 3110
\Fld@onclick@code	3321	\Hy@gttemp	3173, 3175

<code>\Hy@href</code>	267	<code>\HyField@FlagsChoice</code>	2784
<code>\Hy@href@nextactionraw</code>	231	<code>\HyField@FlagsPushButton</code>	2877, 2917, 2968
<code>\Hy@href@page</code>	219	<code>\HyField@FlagsRadioButton</code>	2780
<code>\Hy@linkfileprefix</code>	1542	<code>\HyField@FlagsSubmit</code>	2918
<code>\Hy@linktoc</code>	1553	<code>\HyField@FlagsText</code>	2713
<code>\Hy@MakeCurrentHref</code>	192	<code>\HyField@PDFChoices</code>	2855
<code>\Hy@MakeCurrentHrefAuto</code>	3593, 3609, 3620, 3650, 3664	<code>\HyField@SetKeys</code>	2709, 2750, 2753, 2772, 2870, 2913, 2961, 2996
<code>\Hy@next</code>	3619, 3628, 3631	<code>\HyPat@ObjRef</code>	3521, 3528, 3535, 3542
<code>\Hy@numberline</code>	158, 3549	<code>\hyper@@link</code>	275
<code>\Hy@org@chapter</code>	3617, 3632	<code>\hyper@anchor</code>	672
<code>\Hy@OutlineName</code>	3565	<code>\hyper@anchorend</code>	672, 3584, 3611, 3622, 3652
<code>\Hy@OutlineRerunCheck</code>	3563	<code>\hyper@anchorstart</code>	672, 3584, 3611, 3622, 3652
<code>\Hy@pdfmajorversion</code>	160	<code>\hyper@link</code>	35, 36, 722
<code>\Hy@pdfminorversion</code>	159	<code>\hyper@linkend</code>	35, 36, 766
<code>\Hy@pdfstringtrue</code>	31, 593	<code>\hyper@linkfile</code>	317, 833
<code>\Hy@pdfversion</code>	3168	<code>\hyper@linklaunch</code>	39, 334, 899, 958
<code>\Hy@pstringdef</code>	603, 3148	<code>\hyper@linknamed</code>	41, 168, 964
<code>\Hy@PutCatalog</code>	610	<code>\hyper@linkstart</code>	35, 36, 746
<code>\Hy@raisedlink</code>	3583, 3610, 3621, 3651	<code>\hyper@linkurl</code>	300, 361, 786
<code>\Hy@RestoreLastskip</code>	669	<code>\hyper@normalise</code>	267, 296, 312, 329, 355, 368
<code>\Hy@ReturnAfterFi</code>	12, 3149	<code>\HyPL@Labels</code>	612, 622
<code>\Hy@safe@activestru</code>	806, 2585	<code>\HyPL@storePageLabel</code>	610
<code>\Hy@SaveLastskip</code>	649	<code>\HyPsd@SanitizeForOutFile</code>	3552
<code>\Hy@secnum@part</code>	3638, 3640, 3643	<code>\if@filesw</code>	2565, 2588, 2607
<code>\Hy@SectionAnchorHref</code>	3574, 3598, 3669	<code>\ifFld@combo</code>	2759, 2789
<code>\Hy@SectionHShift</code>	3573, 3582, 3594, 3665	<code>\ifFld@hidden</code>	2714, 2774
<code>\Hy@StepCount</code>	2744, 2810	<code>\ifFld@multiline</code>	2702
<code>\Hy@temp</code>	2586, 2601, 3130, 3131	<code>\ifFld@popdown</code>	2760
<code>\Hy@unicodedefalse</code>	3170	<code>\ifFld@radio</code>	2756, 2779
<code>\Hy@VerboseAnchor</code>	650	<code>\ifHy@implicit</code>	3569
<code>\Hy@VerboseLinkStart</code>	729, 753	<code>\ifHy@pdfescapeform</code>	3116, 3126
<code>\Hy@VerboseLinkStop</code>	737, 774, 816, 884, 944, 988	<code>\ifHy@unicode</code>	3169
<code>\Hy@VersionChecked</code>	151	<code>\kv@set@family@handler</code>	171
<code>\Hy@WrapperDef</code>	646	<code>\m@ne</code>	3625, 3640
<code>\Hy@xspace@end</code>	734, 815, 883, 943, 987	<code>\OBJ@OCG@view</code>	157
<code>\HyAnn@AbsPageLabel</code>	2670, 2676, 2718, 2813, 2857, 2885, 2924, 2971, 3009	<code>\p@</code>	2748
<code>\HyAnn@Count</code>	2674, 2675, 2678, 2679, 2680, 2685, 2689	<code>\pdf@ifdraftmode</code>	2621
<code>\HyField@@AddToFields</code>	2578, 2605	<code>\PDF@SetupDoc</code>	152
<code>\HyField@AddToFields</code>	2576, 2726, 2846, 2863, 2894, 2950, 2979, 3043	<code>\PDFForm@@Name</code>	3156, 3159, 3163, 3166
<code>\HyField@afields</code>	2545	<code>\PDFForm@Check</code>	3010, 3042, 3240
<code>\HyField@AfterAuxOpen</code>	2547, 2573, 2606	<code>\PDFForm@List</code>	2858, 2861, 3326
<code>\HyField@AuxAddToCoFields</code>	2558, 2570, 2591	<code>\PDFForm@Name</code>	2712, 2773, 2874, 3000, 3155
<code>\HyField@AuxAddToFields</code>	2551, 2567, 2609	<code>\PDFForm@Push</code>	2886, 2892, 3292
<code>\HyField@cofields</code>	2546	<code>\PDFForm@Radio</code>	2814, 2834, 3361
<code>\HyField@FlagsCheckBox</code>	3001	<code>\PDFForm@Reset</code>	2972, 2978, 3484
		<code>\PDFForm@Submit</code>	2925, 2944, 3449
		<code>\PDFForm@Text</code>	2719, 2724, 3399
		<code>\protected@edef</code>	698

